User Manual

for S32K1 ICU Driver

Document Number: UM2ICU4.4 Rev0000R1.0.1 Rev. 1.0

Chapter 1

Revision History

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

Chapter 2

Introduction

This User Manual describes NXP Semiconductor AUTOSAR ICU for S32K1. 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 intended to be used with the following microcontroller devices of NXP Semiconductors:

- s32k116_qfn32
- s32k116_lqfp48
- s32k118_lqfp48
- s32k118_lqfp64
- s32k142_lqfp48
- s32k142_lqfp64
- s32k142_lqfp100
- s32k142w_lqfp48
- s32k142w_lqfp64
- s32k144_lqfp48

- s32k144_lqfp64
- s32k144_lqfp100
- s32k144_mapbga100
- s32k144w_lqfp48
- s32k144w_lqfp64
- s32k146_lqfp64
- s32k146_lqfp100
- s32k146_mapbga100
- s32k146_lqfp144
- s32k148_lqfp100
- s32k148_mapbga100
- s32k148_lqfp144
- s32k148_lqfp176

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

2.2 Overview

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

AUTOSAR:

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

2.3 About This Manual

This Technical Reference employs the following typographical conventions:

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

Notes and warnings are shown as below:

Note

This is a note.

Warning

This is a warning

2.4 Acronyms and Definitions

Term	Definition
API	Application Programming Interface
ASM	Assembler
BSMI	Basic Software Make file Interface
CAN	Controller Area Network
C/CPP	C and C++ Source Code
LPCMP	Low Power Comparator
CS	Chip Select
CTU	Cross Trigger Unit
DEM	Diagnostic Event Manager
DET	Development Error Tracer
DMA	Direct Memory Access
ECU	Electronic Control Unit
EMIOS	Enhanced Modular IO Subsystem
FIFO	First In First Out
FTM	Flextimer Module
ICU	Input Capture Unit
ISR	Interrupt Service Routine
LSB	Least Signifigant Bit
MCU	Micro Controller Unit
MIDE	Multi Integrated Development Environment
MSB	Most Significant Bit
N/A	Not Applicable
OS	Operating System
PB Variant	Post Build Variant
PC Variant	Pre Compile Variant
RAM	Random Access Memory
ROM	Read-only Memory
SIUL2	System Integration Unit Lite2
SWS	Software Specification
VLE	Variable Length Encoding
WKPU	Wakeup Unit
XML	Extensible Markup Language

2.5 Reference List

#	Title	Version
1	Specification of ICU Driver	AUTOSAR Release 4.↩ 4.0
2	S32K1xx Series Reference Manual	Rev. 14, 09/2021
3	S32K116_0N96V	Rev. 22/OCT/2021

#	Title	Version
4	S32K118_0N97V	Rev. 22/OCT/2021
5	S32K142_0N33V	Rev. 22/OCT/2021
6	S32K144_0N57U	Rev. 22/OCT/2021
7	S32K144W_0P64A	Rev. 22/OCT/2021
8	S32K146_0N73V	Rev. 22/OCT/2021
9	S32K148_0N20V	Rev. 22/OCT/2021
10	S32K1xx Data Sheet	Rev. 14, 08/2021

Chapter 3

Driver

3.1 Requirements

Requirements for this driver are detailed in the AUTOSAR 4.4 Rev0000ICU Driver Software Specification document (See Table Reference_list).

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

It has vendor-specific requirements and implementation.

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
- Wake-up interrupts

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) can be used for signal measurements. This is because the channel after it (2∗n+1) is used internally by the ICU Driver

The FTM module of S32K1 supports period time measurement, edge detection and notification, edge counting and edge time stamping.

The PORT_CI module of S32K1 supports edge detection and notification.

The LPTMR module of S32K1 supports edge detection and notification, edge counting.

The LPIT module of S32K1 can support edge detection and notification and edge time stamping.

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.

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.

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.

The RTD driver assures reentrancy (single core execution) for the APIs based on the following assumptions:

- the "called-again" API is for a different resource (hardware/logic channel);
- common variables/registers accessed with "rmw" are guarded by Exclusive Areas which need to be correctly implemented in RTE on user side;

3.3 Hardware Resources

The hardware resources configured by the Icu driver are next: **CMP**, **LPIT**, **FTM**, **PORT_CI** and **LPTMR**. The CMP, LPIT, FTM, PORT_CI and LPTMR input signal to microcontroller pin mapping can be done by using "IO_Signal_Description_and_Input_multiplexing_tables.xls" from the Reference manual.

3.4 Deviations from Requirements

Requirement	Status	Description	Notes
SWS_Icu_00150	N/S	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.	Rejection Reason: The requirement is violating safety because:The ICU149 is a safety integrity assumption for external environment, which shall be implemented for FTE; For GTE and NTE I↩ CU149 has a role to increase availablity because the check will be supported by ICU driver; see also 00149
SWS_Icu_00380	N/S	These requirements are not applicable to this specification.	Not a requirement.
SWS_Icu_91002	N/S	Service name: - Icu_Disable↩ NotificationAsync (draft) - Syntax: - void Icu_Disable↩ NotificationAsync(Icu_ChannelType Channel) - Service ID[hex]: - 0x18 - Sync/Async: - Asynchronous - Reentrancy: - Reentrant (limited according to ICU050) - Parameters (in): - Channel - Numeric identifier of the ICU channel. - Parameters (inout): - None - Parameters (out): - None - Return value: - None - Description: - This function disables the notification of a channel.Tags: atp.Status=draft - Available via: - Icu.h -	Description specified as draft is not clear. Should be re-assesed on next ASR version
SWS_Icu_91003	N/S	Service name: - Icu_Enable↩ NotificationAsync (draft) - Syntax: - void Icu_Enable↩ NotificationAsync(Icu_ChannelType Channel) - Service ID[hex]: - 0x19 - Sync/Async: - Asynchronous - Reentrancy: - Non Reentrant Reentrant (limited according to ICU050) - Parameters (in): - Channel - Numeric identifier of the ICU channel. - Parameters (inout): - None - Parameters (out): - None - Return value: - None - Description: - This function enables the notification on the given channel.Tags: atp.Status=draft - Available via: - Icu.h -	Description specified as draft is not clear. Should be re-assesed on next ASR version

3.5 Driver limitations

- Signal measurement with BOTH_EDGE trigger is experimentally supported and returns period and high time only – in analysis to see if speed gain in measurement is valuable.

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. 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 ICU driver has the Dual Clock Feature. The IcuEnableDualClockMode from IcuAutosarExt should be enabled in order to have this feature active. Afterwards, the Prescaler_Alternate 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.
Duty cycle measurement using BOTH EDGES measurement limitation explained.
When this type of measurement is used the limitation consist in considering if the measured pulsWidth is always HIGH TIME.
For example, in the next figure it can be seen that for the same signal, pulseWidth will have the same value, even for the second one measurement where the active time can be considered the LOW TIME as well.
In the I) and the II) case the result will be the same dutyCycle = 25%, even the signal will be consider with LOW TIME as the active time.
Pros for this measurement:
\t -> the measurement will be done with no wait time
Cons for this measurement:
\t -> the result will consider the HIGH TIME as the active time for all the measurements with BOTH EDG↩ES measurement, but there is an workaround if it is known that the active time is the LOW TIME dutyCycle = 100 - currentDutyCycle(e.g in our case if we consider the LOW TIME as the active time dutyCycle=100-25=75%)

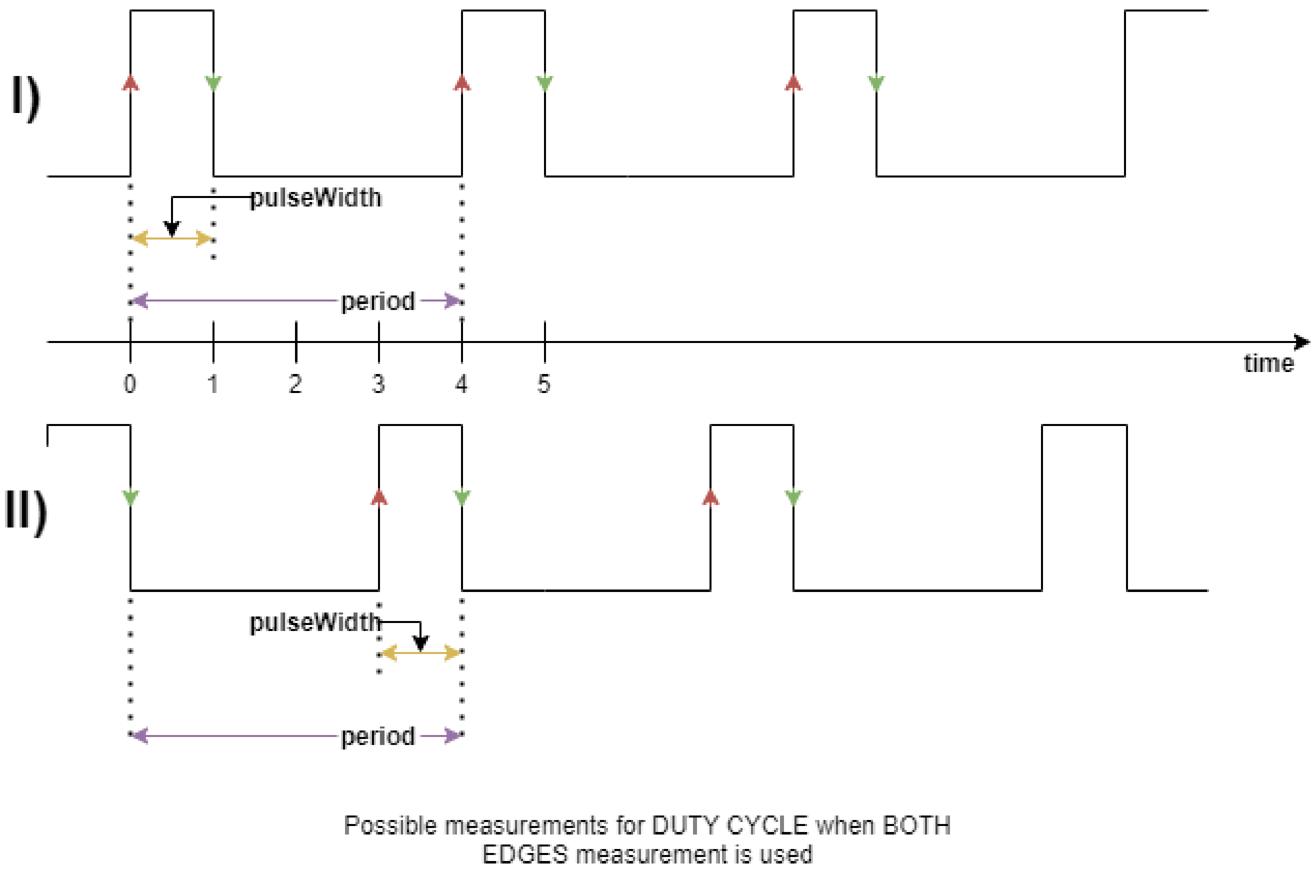
I)
pulseWidth
period
0
1
2
3
4
5
time
II)
pulseWidth
period
Possible measurements for DUTY CYCLE when BOTH EDGES measurement is used

Figure 3.1 Possible measurements for DUTY CYCLE when BOTH EDGES measurement is used

3.7 Runtime Errors

The driver does not generate any DEM runtime errors.

3.8 Symbolic Names Disclaimer

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

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

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

Chapter 4

Tresos Configuration Plug-in

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

- Module Icu
  - Container IcuConfigSet
    - Parameter IcuMaxChannel
    - Container IcuChannel
      - Parameter IcuChannelId
      - Parameter IcuDMAChannelEnable
      - Parameter IcuDefaultStartEdge
      - Parameter IcuMeasurementMode
      - Parameter IcuOverflowNotification
      - Parameter IcuWakeupCapability
      - Reference IcuChannelEcucPartitionRef
      - Reference IcuChannelRef
      - Reference IcuDMAChannelRef
      - Container IcuSignalEdgeDetection
      - Parameter IcuSignalNotification
      - Container IcuSignalMeasurement
      - Parameter IcuSignalMeasurementProperty
      - Container IcuTimestampMeasurement
      - Parameter IcuTimestampMeasurementProperty
      - Parameter IcuTimestampNotification
      - Container IcuWakeup
      - Reference IcuChannelWakeupInfo
    - Container IcuFtm
      - Parameter IcuFtmModule
      - Parameter IcuFtmClockSource
      - Parameter IcuFtmPrescaler
      - Parameter IcuFtmPrescalerAlternate
      - Parameter IcuFtmDebugMode
      - Parameter IcuFtmModValue
      - Container IcuFtmChannels

      - Parameter IcuTrgEnableRoundRobinInterrupt
      - Parameter IcuTrgEnableRoundRobin
    - Container IcuHwInterruptConfigList
      - Parameter IcuIsrHwId
      - Parameter IcuIsrEnable
  - Container IcuGeneral
    - Parameter IcuDevErrorDetect
    - Parameter IcuReportWakeupSource
    - Parameter IcuEnableUserModeSupport
    - Parameter IcuMulticoreSupport
    - Reference IcuEcucPartitionRef
    - Reference IcuKernelEcucPartitionRef
  - Container IcuAutosarExt
    - Parameter IcuEnableDualClockMode
    - Parameter IcuOverflowNotificationApi
    - Parameter IcuGetInputLevelApi
    - Parameter IcuLptmrStandbyWakeupSupport
    - Parameter IcuGetCaptureRegisterValueApi
  - Container IcuOptionalApis
    - Parameter IcuDeInitApi
    - Parameter IcuDisableWakeupApi
    - Parameter IcuEdgeCountApi
    - Parameter IcuEnableWakeupApi
    - Parameter IcuGetDutyCycleValuesApi
    - Parameter IcuGetInputStateApi
    - Parameter IcuGetTimeElapsedApi
    - Parameter IcuGetVersionInfoApi
    - Parameter IcuSetModeApi
    - Parameter IcuSignalMeasurementApi
    - Parameter IcuTimestampApi
    - Parameter IcuWakeupFunctionalityApi
    - Parameter IcuEdgeDetectApi
  - Container CommonPublishedInformation
    - Parameter ArReleaseMajorVersion
    - Parameter ArReleaseMinorVersion
    - Parameter ArReleaseRevisionVersion
    - Parameter ModuleId
    - Parameter SwMajorVersion
    - Parameter SwMinorVersion
    - Parameter SwPatchVersion
    - Parameter VendorApiInfix
    - Parameter VendorId

4.1 Module Icu

Configuration of the Icu (Input Capture Unit) module

Included containers:

- IcuConfigSet
- IcuGeneral
- IcuAutosarExt
- IcuOptionalApis
- CommonPublishedInformation

Property	Value
type	ECUC-MODULE-DEF
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantSupport	true
supportedConfigVariants	VARIANT-POST-BUILD, VARIANT-PRE-COMPILE

4.2 Container IcuConfigSet

This container is the base for a multiple configuration set

Included subcontainers:

- IcuChannel
- IcuFtm
- IcuLpit
- IcuLptmr
- IcuPort
- IcuLpCmp
- IcuHwInterruptConfigList

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

4.3 Parameter IcuMaxChannel

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.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	3
max	75
min	1

4.4 Container IcuChannel

Configuration of an individual ICU channel.

Included subcontainers:

- IcuSignalEdgeDetection
- IcuSignalMeasurement
- IcuTimestampMeasurement
- IcuWakeup

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

4.5 Parameter IcuChannelId

Channel Id of the ICU channel.

This value will be assigned to the symbolic name derived of the IcuChannel container short name.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	true
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	1
max	74
min	0

4.6 Parameter IcuDMAChannelEnable

IcuDMAChannelEnable indicates if the corresponding channel will use DMA for measurement

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

4.7 Parameter IcuDefaultStartEdge

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

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

4.8 Parameter IcuMeasurementMode

Configures the measurement mode of this channel.

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

Implementation Type: Icu_MeasurementModeType

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

4.9 Parameter IcuOverflowNotification

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 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).

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

4.10 Parameter IcuWakeupCapability

Information about the wakeup-capability of this channel.

true: Channel is wakeup capable.

false: Channel is not wakeup capable.

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

4.11 Reference IcuChannelEcucPartitionRef

Maps a ICU channel to zero or multiple ECUC partitions to limit the access to this channel group.

The ECUC partitions referenced are a subset of the ECUC partitions where the ICU driver is mapped to.

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

4.12 Reference IcuChannelRef

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

Property	Value
type	ECUC-CHOICE-REFERENCE-DEF
origin	NXP
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
requiresSymbolicNameValue	False
destinations	['/TS_T40D2M10I1R0/Icu/IcuConfigSet/IcuFtm/IcuFtmChannels', '/TS_T40D2M10I1R0/Icu/IcuConfigSet/IcuLpit/IcuLpitChannels', '/TS_T40D2M10I1R0/Icu/IcuConfigSet/IcuLptmr/IcuLptmrChannels', '/TS_T40D2M10I1R0/Icu/IcuConfigSet/IcuPort/IcuPortChannels', '/TS_T40D2M10I1R0/Icu/IcuConfigSet/IcuLpCmp']

4.13 Reference IcuDMAChannelRef

Icu DMA Channel Reference

Reference to the DMA Channel configure for the Request

Property	Value
type	ECUC-CHOICE-REFERENCE-DEF
origin	NXP
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
requiresSymbolicNameValue	False
destinations	['/TS_T40D2M10I1R0/Mcl/MclConfig/dmaLogicChannel_Type']

4.14 Container IcuSignalEdgeDetection

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

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD

4.15 Parameter IcuSignalNotification

Notification function for signal notification

Property	Value
type	ECUC-FUNCTION-NAME-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	NULL_PTR

4.16 Container IcuSignalMeasurement

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

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.17 Parameter IcuSignalMeasurementProperty

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

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	ICU_DUTY_CYCLE
literals	['ICU_DUTY_CYCLE', 'ICU_HIGH_TIME', 'ICU_LOW_TIME', 'ICU_P↩ ERIOD_TIME']

4.18 Container IcuTimestampMeasurement

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

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0

Property	Value
upperMultiplicity	1
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.19 Parameter IcuTimestampMeasurementProperty

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

Following type of buffer inplemented in current implementation.

ICU_CIRCULAR_BUFFER.

ICU_LINEAR_BUFFER.

Implementation Type: Icu_TimestampBufferType

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

4.20 Parameter IcuTimestampNotification

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

Property	Value
type	ECUC-FUNCTION-NAME-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	0

Property	Value
upperMultiplicity	1
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	NULL_PTR

4.21 Container IcuWakeup

This container contains the configuration (parameters) needed to configure a wakeup capable channel

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.22 Reference IcuChannelWakeupInfo

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

Implementation Type: reference to EcuM_WakeupSourceType

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
postBuildVariantValue	true

Property	Value
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/EcuM/EcuMConfiguration/EcuMCommon↩ Configuration/EcuMWakeupSource

4.23 Container IcuFtm

Configuration of a FTM module available on the platfom.

Included subcontainers:

- IcuFtmChannels

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	8
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.24 Parameter IcuFtmModule

Select which hardware instance of FTM to use.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	7
min	0

4.25 Parameter IcuFtmClockSource

Select origin of clock source used by current instance of FTM.

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

4.26 Parameter IcuFtmPrescaler

Prescaler used by current FTM instance.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	1
max	128
min	0

4.27 Parameter IcuFtmPrescalerAlternate

Set an alternante prescaler for FTM instance. With this option the frequency can be changed at runtime.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	128
max	128
min	0

4.28 Parameter IcuFtmDebugMode

Mode 0: FTM counter stopped, CH(n)F bit can be set, FTM channels in functional mode, writes to MOD,CNTIN and C(n)V registers bypass the register buffers.

Mode 1: FTM counter stopped, CH(n)F bit is not set, FTM channels outputs are forced to their safe value , writes to MOD,CNTIN and C(n)V registers bypass the register buffers.

Mode 2: FTM counter stopped, CH(n)F bit is not set, FTM channels outputs are frozen when chip enters in BDM mode, writes to MOD, CNTIN and C(n)V registers bypass the register buffers.

Mode 3: FTM counter in functional mode, CH(n)F bit can be set,FTM channels in functional mode, writes to MOD,CNTIN and C(n)V registers is in fully functional mode.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	MODE_0
literals	['MODE_0', 'MODE_1', 'MODE_2', 'MODE_3']

4.29 Parameter IcuFtmModValue

Maxim value of counter for current FTM instance. This parameter will define the maxim pulse period which can be measured.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	65535
min	0

4.30 Container IcuFtmChannels

List of Ftm channels available on the platform.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	8
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.31 Parameter IcuFtmChannel

Select a hardware channel of the current FTM instance to configure.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	7
min	0

4.32 Parameter IcuFtmFilter

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.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	15
min	0

4.33 Container IcuLpit

Configuration of a Lpit module available on the platfom.

Included subcontainers:

- IcuLpitChannels

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	2
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE VARIANT-POST-BUILD: PRE-COMPILE

4.34 Parameter IcuLpitModule

Select the physical Lpit Module.

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

4.35 Parameter IcuLpitDebugEnable

Vendor specific: Select to set hardware in debug mode.

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

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

4.36 Container IcuLpitChannels

List of Lpit Channels on flatform.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	4
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.37 Parameter IcuLpitChannel

Selects one of the Lpit channels available on the platform.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	4
min	0

4.38 Parameter IcuLpitTriggerSource

Selects the internal and external trigger sources .

INTERNAL_TRIGGER

EXTERNAL_TRIGGER

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

4.39 Parameter IcuLpitTriggerSelect

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

ICU_LPIT_ICU_TRG_CH0

ICU_LPIT_ICU_TRG_CH1

ICU_LPIT_ICU_TRG_CH2

ICU_LPIT_ICU_TRG_CH3

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	LPIT_ICU_TRG_CH0
literals	['LPIT_ICU_TRG_CH0', 'LPIT_ICU_TRG_CH1', 'LPIT_ICU_TRG_CH2', 'LPIT_ICU_TRG_CH3']

4.40 Container IcuLptmr

Configuration of a Lptmr module available on the platfom.

Included subcontainers:

- IcuLptmrChannels

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	2
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.41 Parameter IcuLptmrModule

Select the physical Lptmr Module.

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

4.42 Parameter PrescalerEnable

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

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

4.43 Parameter IcuLptmrPrescaler

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

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	LPTMR_ICU_GLITCH_FILTER_2
literals	['LPTMR_ICU_GLITCH_FILTER_2', 'LPTMR_ICU_GLITCH_FILTER_4', 'LPTMR_ICU_GLITCH_FILTER_8', 'LPTMR_ICU_GLITCH_FILTER_16', 'LPTMR_ICU_GLITCH_FILTER_32', 'LPTMR_ICU_GLITCH_FILTER_64', 'LPTMR_ICU_GLITCH_FILTER_128', 'LPTMR_ICU_GLITCH_FILTER_256', 'LPTMR_ICU_GLITCH_FILTER_512', 'LPTMR_ICU_GLITCH_FILTER_1024', 'LPTMR_ICU_GLITCH_FILTER_2048', 'LPTMR_ICU_GLITCH_FILTER_4096', 'LPTMR_ICU_GLITCH_FILTER_8192', 'LPTMR_ICU_GLITCH_FILTER_16384', 'LPTMR_ICU_GLITCH_FILTER_32768', 'LPTMR_ICU_GLITCH_FILTER_65536']

4.44 Parameter IcuLptmrChannelClkSrc

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.

LPTMR_ICU_SIRCDIV2_CLK

LPTMR_ICU_LPO1K_CLK

LPTMR_ICU_RTC_CLK

LPTMR_ICU_PCC_LPTMR0

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

Property	Value
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	LPTMR_ICU_SIRCDIV2_CLK
literals	['LPTMR_ICU_SIRCDIV2_CLK', 'LPTMR_ICU_LPO1K_CLK', 'LPTMR↩ _ICU_RTC_CLK', 'LPTMR_ICU_PCC_LPTMR0']

4.45 Parameter IcuLptmrPinSelect

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

ALT1

ALT2

ALT3

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	ALT1
literals	['TRGMUX_OUTPUT', 'ALT1', 'ALT2', 'ALT3']

4.46 Container IcuLptmrChannels

List of Lptmr Channels on flatform.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	2
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.47 Parameter IcuLptmrChannel

Selects one of the Lptmr channels available on the platform.

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

4.48 Container IcuPort

Configuration of a Port module available on the platfom.

Included subcontainers:

- IcuPortChannels

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	5
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.49 Parameter IcuPortModule

Select the physical Port Module.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	4
min	0

4.50 Container IcuPortChannels

List of Port Channels on flatform.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	32
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.51 Parameter IcuPortChannel

Selects one of the Port channels available on the platform.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	31
min	0

4.52 Container IcuLpCmp

Configuration of a LPCMP module available on the platfom.

Included subcontainers:

- IcuCmp
- IcuDac
- IcuTrigger

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	2
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.53 Parameter IcuCmpInstanceNumber

Configure the instance number of IP used.NoteThis is an Implementation Specific Parameter.

Property	Value
type	ECUC-INTEGER-PARAM-DEF

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

4.54 Container IcuCmp

Configuration of a LPCMP module available on the platfom.

Included subcontainers:

- None

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

4.55 Parameter IcuCmpFunctionalMode

Functional mode of LPCMP

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

Property	Value
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	CMP_IP_FUNCTIONALMODE_DISABLED
literals	['CMP_IP_FUNCTIONALMODE_DISABLED', 'CMP_IP_FUNCTIONAL↩MODE_CONTINUOUS', 'CMP_IP_FUNCTIONALMODE_SAMPLED_N↩ONFILTERED_INT_CLK', 'CMP_IP_FUNCTIONALMODE_SAMPLED↩_NONFILTERED_EXT_CLK', 'CMP_IP_FUNCTIONALMODE_SAMPL↩ED_FILTERED_INT_CLK', 'CMP_IP_FUNCTIONALMODE_SAMPLED↩_FILTERED_EXT_CLK', 'CMP_IP_FUNCTIONALMODE_WINDOWED', 'CMP_IP_FUNCTIONALMODE_WINDOWED_RESAMPLED', 'CMP_IP↩_FUNCTIONALMODE_WINDOWED_FILTERED']

4.56 Parameter IcuCmpHysteresisLevel

Internal hysteresis mode of LPCMP - see specific implementation

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	CMP_IP_HYSTERESISLEVEL_0
literals	['CMP_IP_HYSTERESISLEVEL_0', 'CMP_IP_HYSTERESISLEVEL_1', 'CMP_IP_HYSTERESISLEVEL_2', 'CMP_IP_HYSTERESISLEVEL_3']

4.57 Parameter IcuCmpOffsetLevel

Comparator offset control - see specific implementation

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

Property	Value
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	CMP_IP_OFFSETLEVEL_0
literals	['CMP_IP_OFFSETLEVEL_0', 'CMP_IP_OFFSETLEVEL_1']

4.58 Parameter IcuCmpEnablePinOutput

EnablePinOutput.

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

4.59 Parameter IcuCmpEnableInverter

EnableInverter.

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

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

4.60 Parameter IcuCmpEnableComparatorInvert

EnableComparatorInvert.

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

4.61 Parameter IcuCmpEnableHighPowerMode

EnableHighPowerMode.

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

4.62 Parameter IcuCmpFilterSamplePeriod

FilterSamplePeriod

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

4.63 Parameter IcuCmpFilterSampleCount

FilterSampleCount

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	7
min	0

4.64 Parameter IcuCmpEnableDma

EnableDma.

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

4.65 Parameter IcuCmpNegativeInputMux

NegativeInputMux

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	CMP_IP_INPUTMUX_IN0
literals	['CMP_IP_INPUTMUX_IN0', 'CMP_IP_INPUTMUX_IN1', 'CMP_IP_I↩ NPUTMUX_IN2', 'CMP_IP_INPUTMUX_IN3', 'CMP_IP_INPUTMUX_↩ IN4', 'CMP_IP_INPUTMUX_IN5', 'CMP_IP_INPUTMUX_IN6', 'CMP_I↩ P_INPUTMUX_IN7', 'CMP_IP_INPUTMUX_DAC']

4.66 Parameter IcuCmpPositiveInputMux

PositiveInputMux

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF

Property	Value
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	CMP_IP_INPUTMUX_IN0
literals	['CMP_IP_INPUTMUX_IN0', 'CMP_IP_INPUTMUX_IN1', 'CMP_IP_I↩ NPUTMUX_IN2', 'CMP_IP_INPUTMUX_IN3', 'CMP_IP_INPUTMUX_↩ IN4', 'CMP_IP_INPUTMUX_IN5', 'CMP_IP_INPUTMUX_IN6', 'CMP_I↩ P_INPUTMUX_IN7', 'CMP_IP_INPUTMUX_DAC']

4.67 Parameter IcuCmpOutputSelect

OutputSelect

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

4.68 Container IcuDac

Configuration of a LPCMP module available on the platfom.

Included subcontainers:

- None

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

4.69 Parameter IcuDacVoltageLevel

VoltageLevel

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

4.70 Parameter IcuDacVoltageRefSource

VoltageRefSource

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

Property	Value
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	CMP_IP_VOLTAGEREFSOURCE_VREF0
literals	['CMP_IP_VOLTAGEREFSOURCE_VREF0', 'CMP_IP_VOLTAGEREFSOURCE_VREF1']

4.71 Parameter IcuDacPowerState

PowerState

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

4.72 Container IcuTrigger

Configuration of a LPCMP module available on the platfom.

Included subcontainers:

- None

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

4.73 Parameter IcuTrgRoundRobinEnChannelMask

RoundRobinEnChannelMask

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

4.74 Parameter IcuTrgInitDelayValue

InitDelayValue

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	63
min	0

4.75 Parameter IcuTrgSampleDelay

SampleDelay

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	CMP_IP_SAMPLEDELAY_0_CYCLES
literals	['CMP_IP_SAMPLEDELAY_0_CYCLES', 'CMP_IP_SAMPLEDELAY_1_CYCLES', 'CMP_IP_SAMPLEDELAY_2_CYCLES', 'CMP_IP_SAMPLEDELAY_3_CYCLES']

4.76 Parameter IcuTrgFixedChannel

FixedChannel

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	CMP_IP_FIXEDCHANNEL_0
literals	['CMP_IP_FIXEDCHANNEL_0', 'CMP_IP_FIXEDCHANNEL_1', 'CMP_IP_FIXEDCHANNEL_2', 'CMP_IP_FIXEDCHANNEL_3', 'CMP_IP_FIXEDCHANNEL_4', 'CMP_IP_FIXEDCHANNEL_5', 'CMP_IP_FIXEDCHANNEL_6', 'CMP_IP_FIXEDCHANNEL_7']

4.77 Parameter IcuTrgFixedPort

FixedPort

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

4.78 Parameter IcuTrgEnableRoundRobinInterrupt

EnableRoundRobinInterrupt.

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

4.79 Parameter IcuTrgEnableRoundRobin

EnableRoundRobin.

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

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

4.80 Container IcuHwInterruptConfigList

List of HW interrupts available for the entire platform.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	75
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.81 Parameter IcuIsrHwId

Id of the HW interrupt service routine available platform wide and usable by ICU module.

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

Property	Value
defaultValue	FTM_0_CH_1
literals	['FTM_0_CH_0', 'FTM_0_CH_1', 'FTM_0_CH_2', 'FTM_0_CH_3', 'FTM_0_CH_4', 'FTM_0_CH_5', 'FTM_0_CH_6', 'FTM_0_CH_7', 'FTM_1_CH_0', 'FTM_1_CH_1', 'FTM_1_CH_2', 'FTM_1_CH_3', 'FTM_1_CH_4', 'FTM_1_CH_5', 'FTM_1_CH_6', 'FTM_1_CH_7', 'FTM_2_CH_0', 'FTM_2_CH_1', 'FTM_2_CH_2', 'FTM_2_CH_3', 'FTM_2_CH_4', 'FTM_2_CH_5', 'FTM_2_CH_6', 'FTM_2_CH_7', 'FTM_3_CH_0', 'FTM_3_CH_1', 'FTM_3_CH_2', 'FTM_3_CH_3', 'FTM_3_CH_4', 'FTM_3_CH_5', 'FTM_3_CH_6', 'FTM_3_CH_7', 'FTM_4_CH_0', 'FTM_4_CH_1', 'FTM_4_CH_2', 'FTM_4_CH_3', 'FTM_4_CH_4', 'FTM_4_CH_5', 'FTM_4_CH_6', 'FTM_4_CH_7', 'FTM_5_CH_0', 'FTM_5_CH_1', 'FTM_5_CH_2', 'FTM_5_CH_3', 'FTM_5_CH_4', 'FTM_5_CH_5', 'FTM_5_CH_6', 'FTM_5_CH_7', 'FTM_6_CH_0', 'FTM_6_CH_1', 'FTM_6_CH_2', 'FTM_6_CH_3', 'FTM_6_CH_4', 'FTM_6_CH_5', 'FTM_6_CH_6', 'FTM_6_CH_7', 'FTM_7_CH_0', 'FTM_7_CH_1', 'FTM_7_CH_2', 'FTM_7_CH_3', 'FTM_7_CH_4', 'FTM_7_CH_5', 'FTM_7_CH_6', 'FTM_7_CH_7', 'PORT_0', 'PORT_1', 'PORT_2', 'PORT_3', 'PORT_4', 'LPIT_0_CH_0', 'LPIT_0_CH_1', 'LPIT_0_CH_2', 'LPIT_0_CH_3', 'LPTMR_0_CH_0', 'CMP_0']

4.82 Parameter IcuIsrEnable

Status of the HW Interrupt (true - Interrupt shall be enable platform wide; false - Interrupt shall be disabled platform wide.

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

4.83 Container IcuGeneral

Configuration of general ICU parameters.

Included subcontainers:

- None

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

4.84 Parameter IcuDevErrorDetect

Switches the Development Error Detection and Notification on or off.

true: Enabled.

false: Disabled.

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

4.85 Parameter IcuReportWakeupSource

Switch for enabling Wakeup source reporting.

true: Report Wakeup source.

false: Do not report Wakeup source.

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

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

4.86 Parameter IcuEnableUserModeSupport

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

a) configuring REG_PROT for SIUL2 IP so that the registers under protection can be accessed from user mode by setting UAA bit in REG_PROT_GCR to 1

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

for more information, please see chapter 5.7 User Mode Support in IM

Note: Implementation Specific Parameter.

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

4.87 Parameter IcuMulticoreSupport

When this parameter is enabled, the ICU module will adapt to run with Multicore:

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

4.88 Reference IcuEcucPartitionRef

Maps the ICU driver to zero or multiple ECUC partitions to make the driver API available in the according partition.

Depending on the addressed timer resource the interfaces operate as follows.

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

4.89 Reference IcuKernelEcucPartitionRef

Maps the ICU kernel to zero or one ECUC partitions to assign the driver kernel to a certain core.

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

Property	Value
type	ECUC-REFERENCE-DEF

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

4.90 Container IcuAutosarExt

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

Included subcontainers:

- None

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

4.91 Parameter IcuEnableDualClockMode

Enables prescaler settings at mode transition.

true: Enabled.

false: Disabled.

Note: This feature is not required by Autosar.

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

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

4.92 Parameter IcuOverflowNotificationApi

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).

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

4.93 Parameter IcuGetInputLevelApi

Add / removes Icu_GetInputLevel API from the code.

This function returns Input pin state.

Note: This feature is not required by Autosar.

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

4.94 Parameter IcuLptmrStandbyWakeupSupport

Icu_Init() will not clear the IRQ flags (LPTMR) if it is already set during init.

Note: This feature is not required by Autosar and might overule standard.

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

4.95 Parameter IcuGetCaptureRegisterValueApi

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.

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

4.96 Container IcuOptionalApis

This container contains all configuration switches for configuring optional API services of the ICU driver.

Included subcontainers:

- None

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

4.97 Parameter IcuDeInitApi

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

true: Icu_DeInit() can be used.

false: Icu_DeInit() can not be used.

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

4.98 Parameter IcuDisableWakeupApi

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

true: Icu_DisableWakeup() can be used.

false: Icu_DisableWakeup() can not be used.

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

4.99 Parameter IcuEdgeCountApi

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.

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

4.100 Parameter IcuEnableWakeupApi

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

true: Icu_EnableWakeup() can be used.

false: Icu_EnableWakeup() can not be used.

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

4.101 Parameter IcuGetDutyCycleValuesApi

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.

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

4.102 Parameter IcuGetInputStateApi

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

true: Icu_GetInputState() can be used.

false: Icu_GetInputState() can not be used.

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

4.103 Parameter IcuGetTimeElapsedApi

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.

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

4.104 Parameter IcuGetVersionInfoApi

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

true: Icu_GetVersionInfo() can be used.

false: Icu_GetVersionInfo() can not be used.

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

4.105 Parameter IcuSetModeApi

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

true: Icu_SetMode() can be used.

false: Icu_SetMode() can not be used.

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

4.106 Parameter IcuSignalMeasurementApi

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.

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

4.107 Parameter IcuTimestampApi

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.

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

4.108 Parameter IcuWakeupFunctionalityApi

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

true: Icu_CheckWakeup() can be used.

false: Icu_CheckWakeup() can not be used.

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

4.109 Parameter IcuEdgeDetectApi

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.

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

4.110 Container CommonPublishedInformation

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

Included subcontainers:

- None

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

4.111 Parameter ArReleaseMajorVersion

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

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

4.112 Parameter ArReleaseMinorVersion

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

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

4.113 Parameter ArReleaseRevisionVersion

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

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

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

4.114 Parameter ModuleId

Module ID of this module from Module List.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
defaultValue	122
max	122
min	122

4.115 Parameter SwMajorVersion

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

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

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

4.116 Parameter SwMinorVersion

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

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

4.117 Parameter SwPatchVersion

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

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

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

4.118 Parameter VendorApiInfix

In driver modules which can be instantiated several times on a single ECU, BSW00347 requires

that the name of APIs is extended by the VendorId and a vendor specific name.

This parameter is used to specify the vendor specific name. In total, the

implementation specific name is generated as follows: <ModuleName>_>VendorId>_<VendorApiInfix>.

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

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

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

4.119 Parameter VendorId

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

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

None.

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

Chapter 5

Module Index

5.1 Software Specification

Here is a list of all modules:

Chapter 6

Module Documentation

6.1 LPIT IPL

6.1.1 Detailed Description

LPIT HW module.

LPIT IP layer hardware module.

Data Structures

- struct Lpit_Icu_Ip_ChannelConfigType

  LPIT channel configuration structure. More...
- struct Lpit_Icu_Ip_ConfigType

  LPIT IP specific configuration structure type. More...
- struct Lpit_Icu_Ip_ChannelsStateType

  LPIT channels state. More...

Macros

- #define ICU_STOP_SEC_CODE

  LPIT_0 independent ISR declarations.

Types Reference

- typedef void(∗ Lpit_Icu_Ip_NotifyType) (void)

  The notification functions shall have no parameters and no return value.
- typedef void(∗ Lpit_Icu_Ip_CallbackType) (uint16 callbackParam1, boolean callbackParam2)

  Callback type for each channel.
- typedef void(∗ Lpit_Icu_Ip_LogicChState) (uint16 logicChannel, uint8 mask, boolean set)

  Callback type for each channel.

Enum Reference

- enum Lpit_Icu_Ip_StatusType

  Generic error codes.
- enum Lpit_Icu_Ip_MeasurementMode

  LPIT channel measurement mode supported.

Function Reference

- Lpit_Icu_Ip_StatusType Lpit_Icu_Ip_Init (uint8 instance, const Lpit_Icu_Ip_ConfigType ∗userConfig)

  LPIT driver initialization function for LPIT module.
- void Lpit_Icu_Ip_EnableInterrupt (uint8 instance, uint8 hwChannel)

  LPIT driver function that enables interrupts on a LPIT channel.
- void Lpit_Icu_Ip_DisableInterrupt (uint8 instance, uint8 hwChannel)

  LPIT driver function that disables interrupts on a LPIT channel.
- void Lpit_Icu_Ip_EnableNotification (uint8 instance, uint8 hwChannel)

  Driver function Enable Notification for timestamp.
- void Lpit_Icu_Ip_DisableNotification (uint8 instance, uint8 hwChannel)

  Driver function Disable Notification for timestamp.
- uint16 Lpit_Icu_Ip_GetTimestampIndex (uint8 instance, uint8 channel)

  Get timestamp index for timestamp mode.

6.1.2 Data Structure Documentation

6.1.2.1 struct Lpit_Icu_Ip_ChannelConfigType

LPIT channel configuration structure.

Definition at line 129 of file Lpit_Icu_Ip_Types.h.

Data Fields

Type	Name	Description
const uint8	hwChannel	Channel hardware index.
const uint8	triggerSelect	Trigger to use for starting and/or reloading the LPIT timer.
const uint8	triggerSource	Select source of trigger.
Lpit_Icu_Ip_CallbackType	callback	The callback function for channels edge detect events.
uint8	callbackParams	The parameters of callback functions for channels events.
Lpit_Icu_Ip_LogicChState	logicChStateCallback	Store address of function used to change the logic state of the channel in HLD.
const Lpit_Icu_Ip_NotifyType	lpitChannelNotify	The notification functions shall have no parameters and no return value.

6.1.2.2 struct Lpit_Icu_Ip_ConfigType

LPIT IP specific configuration structure type.

Definition at line 144 of file Lpit_Icu_Ip_Types.h.

Data Fields

Type	Name	Description
const uint8	instance	Instance hardware index.
const boolean	debugState	Debug(freeze) mode.
const uint8	numberOfChannels	Number of LPIT channels on the current instance.
const Lpit_Icu_Ip_ChannelConfigType(∗	pChannelsConfig)[ ]	Pointer to the array of configured channels.

6.1.2.3 struct Lpit_Icu_Ip_ChannelsStateType

LPIT channels state.

Definition at line 161 of file Lpit_Icu_Ip_Types.h.

Data Fields

Type	Name	Description
boolean	initState	Initialization status.
Lpit_Icu_Ip_MeasurementMode	measurementMode	Measurement mode for current channel.
Lpit_Icu_Ip_CallbackType	callback	The callback function for channels edge detect events.
uint8	callbackParams	The parameters of callback functions for channels events.
Lpit_Icu_Ip_LogicChState	logicChStateCallback	Store address of function used to change the logic state of the channel in HLD.
Lpit_Icu_Ip_NotifyType	lpitChannelNotify	The notification functions shall have no parameters and no return value.
boolean	notificationEnable	Store the initialization state that determines whether Notifications are enabled, it is always TRUE with standalone IPL and FALSE with AUTOSAR mode.

6.1.3 Macro Definition Documentation

6.1.3.1 ICU_STOP_SEC_CODE

```
#define ICU_STOP_SEC_CODE
```

LPIT_0 independent ISR declarations.

Definition at line 149 of file Lpit_Icu_Ip_Irq.h.

6.1.4 Types Reference

6.1.4.1 Lpit_Icu_Ip_NotifyType

```
typedef void(* Lpit_Icu_Ip_NotifyType) (void)
```

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

Definition at line 100 of file Lpit_Icu_Ip_Types.h.

6.1.4.2 Lpit_Icu_Ip_CallbackType

```
typedef void(* Lpit_Icu_Ip_CallbackType) (uint16 callbackParam1, boolean callbackParam2)
```

Callback type for each channel.

Definition at line 103 of file Lpit_Icu_Ip_Types.h.

6.1.4.3 Lpit_Icu_Ip_LogicChState

```
typedef void(* Lpit_Icu_Ip_LogicChState) (uint16 logicChannel, uint8 mask, boolean set)
```

Callback type for each channel.

Definition at line 106 of file Lpit_Icu_Ip_Types.h.

6.1.5 Enum Reference

6.1.5.1 Lpit_Icu_Ip_StatusType

```
enum Lpit_Icu_Ip_StatusType
```

Generic error codes.

Enumerator

LPIT_IP_STATUS_SUCCESS	Generic operation success status.
LPIT_IP_STATUS_ERROR	Generic operation failure status.

Definition at line 89 of file Lpit_Icu_Ip_Types.h.

6.1.5.2 Lpit_Icu_Ip_MeasurementMode

```
enum Lpit_Icu_Ip_MeasurementMode
```

LPIT channel measurement mode supported.

Enumerator

LPIT_ICU_MODE_NO_MEASUREMENT	No measurement mode.
LPIT_ICU_MODE_SIGNAL_EDGE_DETECT	Signal edge detect measurement mode.
LPIT_ICU_MODE_TIMESTAMP	Timestamp measurement mode.

Definition at line 153 of file Lpit_Icu_Ip_Types.h.

6.1.6 Function Reference

6.1.6.1 Lpit_Icu_Ip_Init()

```
Lpit_Icu_Ip_StatusType Lpit_Icu_Ip_Init (
            uint8 instance,
            const Lpit_Icu_Ip_ConfigType * userConfig )
```

LPIT driver initialization function for LPIT module.

This function is called separately for each LPIT instace and will do the following:

- enables the LPIT module
- configures the debug mode (enabled or disabled)
- disable the IRQ correpsonding to the LPIT channel
- clears the (pending) interrupt flag corresponding to LPIT channel
- enable channel interrupts
- Set Trigger Input Capture Mode

Parameters

in	*instance*	- hardware instance to be configured
in	*userConfig*	- configuration of the instance that will be intialized

Returns

Lpit_Icu_Ip_StatusType

6.1.6.2 Lpit_Icu_Ip_EnableInterrupt()

```
void Lpit_Icu_Ip_EnableInterrupt (
             uint8 instance,
             uint8 hwChannel )
```

LPIT driver function that enables interrupts on a LPIT channel.

This function enables LPIT channel interrupt.

Parameters

in	*instance*	- hardware instance of the module
in	*hwChannel*	- channel instance of the module

6.1.6.3 Lpit_Icu_Ip_DisableInterrupt()

```
void Lpit_Icu_Ip_DisableInterrupt (
             uint8 instance,
             uint8 hwChannel )
```

LPIT driver function that disables interrupts on a LPIT channel.

This function disables LPIT channel interrupt.

Parameters

in	*instance*	- hardware instance of the module
in	*hwChannel*	- channel instance of the module

6.1.6.4 Lpit_Icu_Ip_EnableNotification()

```
void Lpit_Icu_Ip_EnableNotification (
             uint8 instance,
             uint8 hwChannel )
```

Driver function Enable Notification for timestamp.

Parameters

in	*instance*	Hardware instance of FTM used.
in	*hwChannel*	Hardware channel of FTM used.

Returns

void

6.1.6.5 Lpit_Icu_Ip_DisableNotification()

```
void Lpit_Icu_Ip_DisableNotification (
             uint8 instance,
             uint8 hwChannel )
```

Driver function Disable Notification for timestamp.

Parameters

in	*instance*	Hardware instance of FTM used.
in	*hwChannel*	Hardware channel of FTM used.

Returns

void

6.1.6.6 Lpit_Icu_Ip_GetTimestampIndex()

```
uint16 Lpit_Icu_Ip_GetTimestampIndex (
             uint8 instance,
             uint8 channel )
```

Get timestamp index for timestamp mode.

Parameters

in	*instance*	Hardware instance of FTM used.
in	*hwChannel*	Hardware channel of FTM used.

Returns

uint16

6.2 FTM IPL

6.2.1 Detailed Description FTM HW module.

FTM IP layer hardware module.

Data Structures

- struct Ftm_Icu_Ip_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. More...
- struct Ftm_Icu_Ip_ChannelConfigType

  FlexTimer driver Input capture parameters for each channel. More...
- struct Ftm_Icu_Ip_InstanceConfigType

  FTM IP layer module configuration. More...
- struct Ftm_Icu_Ip_ConfigType

  FTM driver input capture parameters. More...
- struct Ftm_Icu_Ip_ChStateType

  This structure is used by the IPL driver for internal logic. More...
- struct Ftm_Icu_Ip_InstStateType

  This structure is used by the IPL driver for internal logic. More...

Macros

- #define CHAN0_IDX

  Channel number for CHAN0.
- #define CHAN1_IDX

  Channel number for CHAN1.
- #define CHAN2_IDX

  Channel number for CHAN2.
- #define CHAN3_IDX

  Channel number for CHAN3.
- #define CHAN4_IDX

  Channel number for CHAN4.
- #define FTM_MAX_VAL_COUNTER

  Maximum value of FTM counter.
- #define FTM_COMBINE_COMBINEx_SHIFT(u8ChannelIdx)

  Set flag COMINEx for specified pair: 0, 1, 2.

Types Reference

- typedef void(∗ Ftm_Icu_Ip_NotifyType) (void)

  The notification functions shall have no parameters and no return value.
- typedef void(∗ Ftm_Icu_Ip_CallbackType) (uint16 callbackParam1, boolean callbackParam2)

  Callback type for each channel.
- typedef void(∗ Ftm_Icu_Ip_LogicChState) (uint16 logicChannel, uint8 mask, boolean set)

  Callback type for each channel.

Enum Reference

- enum Ftm_Icu_Ip_ClockSourceType

  FTM clock source selection.
- enum Ftm_Icu_Ip_DebugModeType

  FTM debug modes of operation.
- enum Ftm_Icu_Ip_EdgeType

  Activation condition for the measurement - selecting edge type.
- enum Ftm_Icu_Ip_ModeType

  Operation mode for ICU driver.
- enum Ftm_Icu_Ip_SubModeType

  Enable/disable DMA support for timestamp.
- enum Ftm_Icu_Ip_MeasType

  Type of operation for signal measurement.
- enum Ftm_Icu_Ip_ClockModeType

  Definition of prescaler type (Normal or Alternate)
- enum Ftm_Icu_Ip_StatusType

  Generic error codes.

Function Reference

- Ftm_Icu_Ip_StatusType Ftm_Icu_Ip_Init (uint8 instance, const Ftm_Icu_Ip_ConfigType ∗userConfig)
- void Ftm_Icu_Ip_SetActivationCondition (uint8 instance, uint8 hwChannel, Ftm_Icu_Ip_EdgeType activation)
- void Ftm_Icu_Ip_EnableInterrupt (uint8 instance, uint8 hwChannel)

  Enable channel interrupt.
- void Ftm_Icu_Ip_DisableInterrupt (uint8 instance, uint8 hwChannel)

  Disable channel interrupt.
- void Ftm_Icu_Ip_EnableNotification (uint8 instance, uint8 hwChannel)

  Driver function Enable Notification for timestamp.
- void Ftm_Icu_Ip_DisableNotification (uint8 instance, uint8 hwChannel)

  Driver function Disable Notification for timestamp.

6.2.2 Data Structure Documentation

6.2.2.1 struct Ftm_Icu_Ip_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.

Definition at line 251 of file Ftm_Icu_Ip_Types.h.

Data Fields

Type	Name	Description
uint16	ActiveTime	Low or High time value.
uint16	PeriodTime	Period time value.

6.2.2.2 struct Ftm_Icu_Ip_ChannelConfigType

FlexTimer driver Input capture parameters for each channel.

Definition at line 264 of file Ftm_Icu_Ip_Types.h.

Data Fields

Type	Name	Description
uint8	hwChannel	Physical hardware channel ID
Ftm_Icu_Ip_ModeType	chMode	FlexTimer module mode of operation
Ftm_Icu_Ip_SubModeType	chSubMode	FlexTimer specific name of operation to execute
Ftm_Icu_Ip_MeasType	measurementMode	Measurement Mode for signal measurement
Ftm_Icu_Ip_EdgeType	edgeAlignement	Edge alignment Mode for signal measurement
boolean	continuouseEn	Continuous measurement state
uint8	filterValue	Filter Value
Ftm_Icu_Ip_CallbackType	callback	The callback function for channels edge detect events
uint8	callbackParams	The parameters of callback functions for channels events
Ftm_Icu_Ip_LogicChState	logicChStateCallback	Store address of function used to change the logic state of the channel in HLD.
Ftm_Icu_Ip_NotifyType	ftmChannelNotification	The notification functions shall have no parameters and no return value.
Ftm_Icu_Ip_NotifyType	ftmOverflowNotification	The overflow notification functions shall have no parameters and no return value.

6.2.2.3 struct Ftm_Icu_Ip_InstanceConfigType

FTM IP layer module configuration.

Definition at line 284 of file Ftm_Icu_Ip_Types.h.

Data Fields

Type	Name	Description
Ftm_Icu_Ip_ClockSourceType	cfgClkSrc	Type of clock source used.
uint8	cfgPrescaler	Prescaler value.
Ftm_Icu_Ip_DebugModeType	debugMode	Debug mode.
uint16	maxCountValue	Maximum counter value. Minimum value is 0.

6.2.2.4 struct Ftm_Icu_Ip_ConfigType

FTM driver input capture parameters.

Definition at line 301 of file Ftm_Icu_Ip_Types.h.

Data Fields

Type	Name	Description
uint8	nNumChannels	Number of input capture channel used.
const Ftm_Icu_Ip_InstanceConfigType ∗	pInstanceConfig	Input capture instance configuration.
const Ftm_Icu_Ip_ChannelConfigType(∗	pChannelsConfig)[ ]	Input capture channels configuration.

6.2.2.5 struct Ftm_Icu_Ip_ChStateType

This structure is used by the IPL driver for internal logic.

Definition at line 313 of file Ftm_Icu_Ip_Types.h.

Data Fields

Type	Name	Description
Ftm_Icu_Ip_NotifyType	ftmChannelNotification	The notification functions for TIME_STAMP or SIGNAL_EDGE_DETECT mode.
Ftm_Icu_Ip_NotifyType	ftmOverflowNotification	The overflow notification functions.
Ftm_Icu_Ip_ModeType	channelMode	FTM channel mode.
Ftm_Icu_Ip_EdgeType	edgeTrigger	Type of edge used for activation.
Ftm_Icu_Ip_SubModeType	dmaMode	Support DMA or not.
Ftm_Icu_Ip_LogicChState	logicChStateCallback	Store address of function used to change the logic state of the channel in HLD.
Ftm_Icu_Ip_CallbackType	callback	Calback for other types of measurement.
uint16	callbackParam	Logic channel for which callback is executed.
boolean	notificationEnable	Store the state determines whether Notifications are enabled or not.

6.2.2.6 struct Ftm_Icu_Ip_InstStateType

This structure is used by the IPL driver for internal logic.

Definition at line 365 of file Ftm_Icu_Ip_Types.h.

Data Fields

Type	Name	Description
boolean	instInit	Module initialization state.

Data Fields

Type	Name	Description
uint8	prescaler	Module prescaler value.
uint8	prescalerAlternate	Module alternate prescaler value.
uint8	spuriousMask	

6.2.3 Macro Definition Documentation

6.2.3.1 CHAN0_IDX

```
#define CHAN0_IDX
```

Channel number for CHAN0.

Definition at line 98 of file Ftm_Icu_Ip_Types.h.

6.2.3.2 CHAN1_IDX

```
#define CHAN1_IDX
```

Channel number for CHAN1.

Definition at line 100 of file Ftm_Icu_Ip_Types.h.

6.2.3.3 CHAN2_IDX

```
#define CHAN2_IDX
```

Channel number for CHAN2.

Definition at line 102 of file Ftm_Icu_Ip_Types.h.

6.2.3.4 CHAN3_IDX

```
#define CHAN3_IDX
```

Channel number for CHAN3.

Definition at line 104 of file Ftm_Icu_Ip_Types.h.

6.2.3.5 CHAN4_IDX

```
#define CHAN4_IDX
```

Channel number for CHAN4.

Definition at line 106 of file Ftm_Icu_Ip_Types.h.

6.2.3.6 FTM_MAX_VAL_COUNTER

```
#define FTM_MAX_VAL_COUNTER
```

Maximum value of FTM counter.

Definition at line 109 of file Ftm_Icu_Ip_Types.h.

6.2.3.7 FTM_COMBINE_COMBINEx_SHIFT

```
#define FTM_COMBINE_COMBINEx_SHIFT(
              u8ChannelIdx )
```

Set flag COMINEx for specified pair: 0, 1, 2.

Definition at line 113 of file Ftm_Icu_Ip_Types.h.

6.2.4 Types Reference

6.2.4.1 Ftm_Icu_Ip_NotifyType

```
typedef void(* Ftm_Icu_Ip_NotifyType) (void)
```

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

Definition at line 246 of file Ftm_Icu_Ip_Types.h.

6.2.4.2 Ftm_Icu_Ip_CallbackType

```
typedef void(* Ftm_Icu_Ip_CallbackType) (uint16 callbackParam1, boolean callbackParam2)
```

Callback type for each channel.

Definition at line 258 of file Ftm_Icu_Ip_Types.h.

6.2.4.3 Ftm_Icu_Ip_LogicChState

```
typedef void(* Ftm_Icu_Ip_LogicChState) (uint16 logicChannel, uint8 mask, boolean set)
```

Callback type for each channel.

Definition at line 261 of file Ftm_Icu_Ip_Types.h.

6.2.5 Enum Reference

6.2.5.1 Ftm_Icu_Ip_ClockSourceType

```
enum Ftm_Icu_Ip_ClockSourceType
```

FTM clock source selection.

Enumerator

FTM_NO_CLOCK_SELECTED	No clock selected. This in effect disables the FTM counter.
FTM_SYSTEM_CLOCK	FTM input clock.
FTM_FIXED_FREQUENCY_CLOCK	Fixed frequency clock.
FTM_EXTERNAL_CLOCK	External clock.

Definition at line 121 of file Ftm_Icu_Ip_Types.h.

6.2.5.2 Ftm_Icu_Ip_DebugModeType

```
enum Ftm_Icu_Ip_DebugModeType
```

FTM debug modes of operation.

Enumerator

MODE↩ _0	FTM counter - stopped.
MODE↩ _1	FTM counter - stopped.
MODE↩ _2	FTM counter - stopped.
MODE↩ _3	FTM counter - functional mode.

Definition at line 134 of file Ftm_Icu_Ip_Types.h.

6.2.5.3 Ftm_Icu_Ip_EdgeType

```
enum Ftm_Icu_Ip_EdgeType
```

Activation condition for the measurement - selecting edge type.

Enumerator

FTM_ICU_NO_PIN_CONTROL	No trigger.
FTM_ICU_RISING_EDGE	Rising edge trigger.
FTM_ICU_FALLING_EDGE	Falling edge trigger.
FTM_ICU_BOTH_EDGES	Rising and falling edge trigger.

Definition at line 147 of file Ftm_Icu_Ip_Types.h.

6.2.5.4 Ftm_Icu_Ip_ModeType

```
enum Ftm_Icu_Ip_ModeType
```

Operation mode for ICU driver.

Enumerator

FTM_ICU_MODE_NO_MEASUREMENT	No measurement mode.
FTM_ICU_MODE_SIGNAL_EDGE_DETECT	Signal edge detect measurement mode.
FTM_ICU_MODE_SIGNAL_MEASUREMENT	Signal measurement mode.
FTM_ICU_MODE_TIMESTAMP	Timestamp measurement mode.
FTM_ICU_MODE_EDGE_COUNTER	Edge counter measurement mode.

Definition at line 160 of file Ftm_Icu_Ip_Types.h.

6.2.5.5 Ftm_Icu_Ip_SubModeType

```
enum Ftm_Icu_Ip_SubModeType
```

Enable/disable DMA support for timestamp.

Definition at line 175 of file Ftm_Icu_Ip_Types.h.

6.2.5.6 Ftm_Icu_Ip_MeasType

```
enum Ftm_Icu_Ip_MeasType
```

Type of operation for signal measurement.

Enumerator

FTM_ICU_NO_MEASUREMENT	No measurement.
FTM_ICU_LOW_TIME	The time measurement for OFF period.
FTM_ICU_HIGH_TIME	The time measurement for ON period.
FTM_ICU_PERIOD_TIME	Period measurement between two consecutive falling/raising edges.
FTM_ICU_DUTY_CYCLE	The fraction of active period.

Definition at line 184 of file Ftm_Icu_Ip_Types.h.

6.2.5.7 Ftm_Icu_Ip_ClockModeType

```
enum Ftm_Icu_Ip_ClockModeType
```

Definition of prescaler type (Normal or Alternate)

Enumerator

FTM_ICU_NORMAL_CLK	Normal prescaler
FTM_ICU_ALTERNATE_CLK	Alternate prescaler

Definition at line 212 of file Ftm_Icu_Ip_Types.h.

6.2.5.8 Ftm_Icu_Ip_StatusType

```
enum Ftm_Icu_Ip_StatusType
```

Generic error codes.

Enumerator

FTM_IP_STATUS_SUCCESS	Generic operation success status.
FTM_IP_STATUS_ERROR	Generic operation failure status.

Definition at line 220 of file Ftm_Icu_Ip_Types.h.

6.2.6 Function Reference

6.2.6.1 Ftm_Icu_Ip_Init()

```
Ftm_Icu_Ip_StatusType Ftm_Icu_Ip_Init (
            uint8 instance,
            const Ftm_Icu_Ip_ConfigType * userConfig )
```

This function configures the channel in the Input Capture mode for either getting time-stamps on edge detection or on signal measurement. When the edge specified in the captureMode argument occurs on the channel and then the FTM counter is captured into the CnV register. The user have to read the CnV register separately to get this value. The filter function is disabled if the filterVal argument passed as 0. The filter feature. is available only on channels 0,1,2,3.

Parameters

in	*instance*	Hardware instance of FTM used.
in	*userConfig*	Configuration of the input capture channel.

Returns

Ftm_Icu_Ip_StatusType

- FTM_IP_STATUS_SUCCESS : Completed successfully.
- FTM_IP_STATUS_ERROR : Error occurred.

6.2.6.2 Ftm_Icu_Ip_SetActivationCondition()

```
void Ftm_Icu_Ip_SetActivationCondition (
            uint8 instance,
            uint8 hwChannel,
            Ftm_Icu_Ip_EdgeType activation )
```

This function enables the requested activation condition(rising, falling or both edges) for corresponding FTM channels.

Parameters

in	*instance*	Hardware instance of FTM used.
in	*hwChannel*	Hardware channel of FTM used.
in	*activation*	Edge activation type used.

Returns

void

6.2.6.3 Ftm_Icu_Ip_EnableInterrupt()

```
void Ftm_Icu_Ip_EnableInterrupt (
            uint8 instance,
            uint8 hwChannel )
```

Enable channel interrupt.

Parameters

in	*instance*	Hardware instance of FTM used.
in	*hwChannel*	Hardware channel of FTM used.

Returns

void.

6.2.6.4 Ftm_Icu_Ip_DisableInterrupt()

```
void Ftm_Icu_Ip_DisableInterrupt (
            uint8 instance,
            uint8 hwChannel )
```

Disable channel interrupt.

Parameters

in	*instance*	Hardware instance of FTM used.
in	*hwChannel*	Hardware channel of FTM used.

Returns

void.

6.2.6.5 Ftm_Icu_Ip_EnableNotification()

```
void Ftm_Icu_Ip_EnableNotification (
             uint8 instance,
             uint8 hwChannel )
```

Driver function Enable Notification for timestamp.

Parameters

in	*instance*	Hardware instance of FTM used.
in	*hwChannel*	Hardware channel of FTM used.

Returns

void

6.2.6.6 Ftm_Icu_Ip_DisableNotification()

```
void Ftm_Icu_Ip_DisableNotification (
             uint8 instance,
             uint8 hwChannel )
```

Driver function Disable Notification for timestamp.

Parameters

in	*instance*	Hardware instance of FTM used.
in	*hwChannel*	Hardware channel of FTM used.

Returns

void

6.3 LPTMR IPL

6.3.1 Detailed Description LPTMR HW module.

LPTMR IP layer hardware module.

Data Structures

- struct Lptmr_Icu_Ip_ChannelConfigType

  Lptmr Channel specific configuration structure type. More...
- struct Lptmr_Icu_Ip_ConfigType

  Lptmr IP specific configuration structure type. More...

Macros

- #define ICU_STOP_SEC_CODE

  LPTRM_0 Channel 0 independent ISR declarations.
- #define LPTMR_ICU_IP_TM_MODE

  Pulse Time Counter mode.

Types Reference

- typedef void(∗ Lptmr_Icu_Ip_NotifyType) (void)

  The notification functions shall have no parameters and no return value.
- typedef void(∗ Lptmr_Icu_Ip_CallbackType) (uint16 logicChannel, boolean overflow)

  Callback type for each channel.

Enum Reference

- enum Lptmr_Icu_PrescalerType

  Prescaler Selection.
- enum Lptmr_Icu_MeasurementModeType

  Lptmr_Icu_ChannelMeasurementModeType.
- enum Lptmr_Icu_Ip_PinSelectType

  Definition of input pin type.
- enum Lptmr_Icu_Ip_ClockSourceType

  LPTMR clock source selection.
- enum Lptmr_Icu_Ip_EdgeType

  Activation condition for the measurement - selecting edge type.
- enum Lptmr_Icu_Ip_StatusType

  Generic error codes.

Function Reference

- Lptmr_Icu_Ip_StatusType Lptmr_Icu_Ip_Init (uint8 instance, const Lptmr_Icu_Ip_ConfigType *user←
Config)
- void Lptmr_Icu_Ip_SetActivationCondition (uint8 instance, Lptmr_Icu_Ip_EdgeType activation)

  This function enables the requested activation condition(rising, falling or both edges) for corresponding LPTMR channels.
- void Lptmr_Icu_Ip_EnableNotification (uint8 instance)

  Driver function Enable Notification for timestamp.
- void Lptmr_Icu_Ip_DisableNotification (uint8 instance)

  Driver function Disable Notification for timestamp.

6.3.2 Data Structure Documentation

6.3.2.1 struct Lptmr_Icu_Ip_ChannelConfigType

Lptmr Channel specific configuration structure type.

Definition at line 198 of file Lptmr_Icu_Ip_Types.h.

Data Fields

Type	Name	Description
uint8	HwChannel	Physical hardware channel ID
const Lptmr_Icu_Ip_EdgeType	DefaultStartEdge	Lptmr Default Start Edge
const Lptmr_Icu_MeasurementModeType	MeasurementModeType	Lptmr MeasurementMode
Lptmr_Icu_Ip_NotifyType	lptmrChannelNotification	The notification functions shall have no parameters and no return value.
Lptmr_Icu_Ip_CallbackType	callback	interrupt callback function.
uint16	callbackParam	The parameters of callback functions for channels events

6.3.2.2 struct Lptmr_Icu_Ip_ConfigType

Lptmr IP specific configuration structure type.

Definition at line 211 of file Lptmr_Icu_Ip_Types.h.

Data Fields

Type	Name	Description
uint8	nNumChannels	Number of Lptmr channels in the Icu configuration
const Lptmr_Icu_Ip_PinSelectType	PinSelect	Lptmr channel parameters
const Lptmr_Icu_Ip_ClockSourceType	ClockSource	Lptmr clock source
const Lptmr_Icu_PrescalerType	Prescaler	The Lptmr Prescaler values
boolean	PrescalerEnable	The Lptmr Prescaler Bypass
const Lptmr_Icu_Ip_ChannelConfigType(∗	pChannelsConfig)[ ]	Pointer to the configured channels for Lptmr

6.3.3 Macro Definition Documentation

6.3.3.1 ICU_STOP_SEC_CODE

```
#define ICU_STOP_SEC_CODE
```

LPTRM_0 Channel 0 independent ISR declarations.

Definition at line 140 of file Lptmr_Icu_Ip_Irq.h.

6.3.3.2 LPTMR_ICU_IP_TM_MODE

```
#define LPTMR_ICU_IP_TM_MODE
```

Pulse Time Counter mode.

Definition at line 90 of file Lptmr_Icu_Ip_Types.h.

6.3.4 Types Reference

6.3.4.1 Lptmr_Icu_Ip_NotifyType

```
typedef void(∗ Lptmr_Icu_Ip_NotifyType) (void)
```

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

Definition at line 177 of file Lptmr_Icu_Ip_Types.h.

6.3.4.2 Lptmr_Icu_Ip_CallbackType

```
typedef void(* Lptmr_Icu_Ip_CallbackType) (uint16 logicChannel, boolean overflow)
```

Callback type for each channel.

Definition at line 180 of file Lptmr_Icu_Ip_Types.h.

6.3.5 Enum Reference

6.3.5.1 Lptmr_Icu_PrescalerType

```
enum Lptmr_Icu_PrescalerType
```

Prescaler Selection.

Enumerator

LPTMR_ICU_GLITCH_FILTER_2	Timer mode: prescaler 2, Glitch filter mode: invalid
LPTMR_ICU_GLITCH_FILTER_4	Timer mode: prescaler 4, Glitch filter mode: 2 clocks
LPTMR_ICU_GLITCH_FILTER_8	Timer mode: prescaler 8, Glitch filter mode: 4 clocks
LPTMR_ICU_GLITCH_FILTER_16	Timer mode: prescaler 16, Glitch filter mode: 8 clocks
LPTMR_ICU_GLITCH_FILTER_32	Timer mode: prescaler 32, Glitch filter mode: 16 clocks
LPTMR_ICU_GLITCH_FILTER_64	Timer mode: prescaler 64, Glitch filter mode: 32 clocks
LPTMR_ICU_GLITCH_FILTER_128	Timer mode: prescaler 128, Glitch filter mode: 64 clocks
LPTMR_ICU_GLITCH_FILTER_256	Timer mode: prescaler 256, Glitch filter mode: 128 clocks
LPTMR_ICU_GLITCH_FILTER_512	Timer mode: prescaler 512, Glitch filter mode: 256 clocks
LPTMR_ICU_GLITCH_FILTER_1024	Timer mode: prescaler 1024, Glitch filter mode: 512 clocks
LPTMR_ICU_GLITCH_FILTER_2048	Timer mode: prescaler 2048, Glitch filter mode: 1024 clocks
LPTMR_ICU_GLITCH_FILTER_4096	Timer mode: prescaler 4096, Glitch filter mode: 2048 clocks
LPTMR_ICU_GLITCH_FILTER_8192	Timer mode: prescaler 8192, Glitch filter mode: 4096 clocks
LPTMR_ICU_GLITCH_FILTER_16384	Timer mode: prescaler 16384, Glitch filter mode: 8192 clocks
LPTMR_ICU_GLITCH_FILTER_32768	Timer mode: prescaler 32768, Glitch filter mode: 16384 clocks
LPTMR_ICU_GLITCH_FILTER_65536	Timer mode: prescaler 65536, Glitch filter mode: 32768 clocks

Definition at line 98 of file Lptmr_Icu_Ip_Types.h.

6.3.5.2 Lptmr_Icu_MeasurementModeType

```
enum Lptmr_Icu_MeasurementModeType
```

Lptmr_Icu_ChannelMeasurementModeType.

Type that indicates the channel mode type(capture mode, edge counter).

Definition at line 122 of file Lptmr_Icu_Ip_Types.h.

6.3.5.3 Lptmr_Icu_Ip_PinSelectType

```
enum Lptmr_Icu_Ip_PinSelectType
```

Definition of input pin type.

Enumerator

TRGMUX_OUTPUT	TRGMUX output.
ALT1	LPTMR_ALT1 pin.
ALT2	LPTMR_ALT2 pin.
ALT3	LPTMR_ALT3 pin.

Definition at line 133 of file Lptmr_Icu_Ip_Types.h.

6.3.5.4 Lptmr_Icu_Ip_ClockSourceType

```
enum Lptmr_Icu_Ip_ClockSourceType
```

LPTMR clock source selection.

Enumerator

LPTMR_ICU_SIRCDIV2_CLK	Select SIRCDIV2_CLK.
LPTMR_ICU_LPO1K_CLK	Select LPO1K_CLK.
LPTMR_ICU_RTC_CLK	Select RTC_CLK.
LPTMR_ICU_PCC_LPTMR0	Select PCC_LPTMR0.

Definition at line 146 of file Lptmr_Icu_Ip_Types.h.

6.3.5.5 Lptmr_Icu_Ip_EdgeType

```
enum Lptmr_Icu_Ip_EdgeType
```

Activation condition for the measurement - selecting edge type.

Enumerator

LPTMR_ICU_RISING_EDGE	Rising edge trigger.
LPTMR_ICU_FALLING_EDGE	Rising edge trigger.

Definition at line 157 of file Lptmr_Icu_Ip_Types.h.

6.3.5.6 Lptmr_Icu_Ip_StatusType

```
enum Lptmr_Icu_Ip_StatusType
```

Generic error codes.

Enumerator

LPTMR_IP_STATUS_SUCCESS	Generic operation success status.
LPTMR_IP_STATUS_ERROR	Generic operation failure status.

Definition at line 166 of file Lptmr_Icu_Ip_Types.h.

6.3.6 Function Reference

6.3.6.1 Lptmr_Icu_Ip_Init()

```
Lptmr_Icu_Ip_StatusType Lptmr_Icu_Ip_Init (
             uint8 instance,
             const Lptmr_Icu_Ip_ConfigType * userConfig )
```

Parameters

in	*instance*	- Hardware instance of LPTMR used.
in	*param*	- Configuration of the input capture channel.

Returns

void

6.3.6.2 Lptmr_Icu_Ip_SetActivationCondition()

```
void Lptmr_Icu_Ip_SetActivationCondition (
            uint8 instance,
            Lptmr_Icu_Ip_EdgeType activation )
```

This function enables the requested activation condition(rising, falling or both edges) for corresponding LPTMR channels.

Parameters

in	*instance*	- Hardware instance of LPTMR used.
in	*activation*	Edge activation type used. • LPTMR_ICU_RISING_EDGE : count pulse on Rising Edge • LPTMR_ICU_FALLING_EDGE: count pulse on Falling Edge

Returns

void

6.3.6.3 Lptmr_Icu_Ip_EnableNotification()

```
void Lptmr_Icu_Ip_EnableNotification (
            uint8 instance )
```

Driver function Enable Notification for timestamp.

6.3.6.4 Lptmr_Icu_Ip_DisableNotification()

```
void Lptmr_Icu_Ip_DisableNotification (
            uint8 instance )
```

Driver function Disable Notification for timestamp.

6.4 PORT_CI IPL

6.4.1 Detailed Description

PORT_CI HW module.

PORT_CI IP layer hardware module.

PORT_CI provides control over all electrical pin controls and ports with 16 bits of bidirectional, general-purpose input and output signals. PORT_CI enables you to select the functions and electrical characteristics that appear on external chip pins. It also controls the multiplexing of internal signals from one module to another and controls chip I/O. It supports as many as 32 external interrupts with trigger event configuration.

PORT_CI provides dedicated pad control to general-purpose pads that can be configured as either inputs or outputs. It provides registers for you to read values from GPIO pads configured as inputs and to write values to GPIO pads configured as outputs:

- When configured as output, you can write to an internal register to control the state driven on the associated output pad.
- When configured as **input**, you can detect the state of the associated pad by reading the value from an internal register.
- When configured as input and output, the pad value can be read back to check if the written value appeared on the pad.

Data Structures

- struct Port_Ci_Icu_Ip_ConfigType

  Port_Ci IP specific configuration structure type. More...
- struct Port_Ci_Icu_Ip_State

  PORT CI IP state structure. More...

Types Reference

- typedef void(∗ Port_Ci_Icu_Ip_NotifyType) (void)

  The notification functions shall have no parameters and no return value.
- typedef void(∗ Port_Ci_Icu_Ip_CallbackType) (uint16 callbackParam1, boolean callbackParam2)

  Callback signature used in each channel with an active interrupt.

Enum Reference

- enum Port_Ci_Icu_Ip_EdgeType

  Port_Ci_Icu_Ip_EdgeType.
- enum Port_Ci_Icu_Ip_StatusType

  Generic error codes.

Function Reference

- void Port_Ci_Icu_Ip_EnableNotification (uint8 instance, uint8 hwChannel)

  Driver function Enable Notification for timestamp.
- void Port_Ci_Icu_Ip_DisableNotification (uint8 instance, uint8 hwChannel)

  Driver function Disable Notification for timestamp.

6.4.2 Data Structure Documentation

6.4.2.1 struct Port_Ci_Icu_Ip_ConfigType

Port_Ci IP specific configuration structure type.

Definition at line 150 of file Port_Ci_Icu_Ip_Types.h.

Data Fields

Type	Name	Description
uint8	nNumChannels	
const Port_Ci_Icu_Ip_ChannelConfigType(∗	pChannelsConfig)[ ]	Number of Port_Ci channels in the Icu configuration.

6.4.2.2 struct Port_Ci_Icu_Ip_State

PORT CI IP state structure.

This structure is used by the IPL driver for internal logic. The content is populated at initialization time.

Definition at line 161 of file Port_Ci_Icu_Ip_Types.h.

Data Fields

Type	Name	Description
boolean	chInit	
Port_Ci_Icu_Ip_CallbackType	callback	Initialization state.
Port_Ci_Icu_Ip_NotifyType	PortCiChannelNotification	Pointer to the callback function.
uint16	callbackParam	The notification functions for SIGNAL_EDGE_DETECT mode.
boolean	notificationEnable	The logic channel for which callback is set.

6.4.3 Types Reference

6.4.3.1 Port_Ci_Icu_Ip_NotifyType

```
typedef void(* Port_Ci_Icu_Ip_NotifyType) (void)
```

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

Definition at line 134 of file Port_Ci_Icu_Ip_Types.h.

6.4.3.2 Port_Ci_Icu_Ip_CallbackType

```
typedef void(* Port_Ci_Icu_Ip_CallbackType) (uint16 callbackParam1, boolean callbackParam2)
```

Callback signature used in each channel with an active interrupt.

Definition at line 136 of file Port_Ci_Icu_Ip_Types.h.

6.4.4 Enum Reference

6.4.4.1 Port_Ci_Icu_Ip_EdgeType

```
enum Port_Ci_Icu_Ip_EdgeType
```

Port_Ci_Icu_Ip_EdgeType.

This indicates the activation type Port_Ci channel (Rising, Falling or Both)

Enumerator

PORT_CI_ICU_FALLING_EDGE	No trigger. An appropriate action shall be executed when a falling edge occurs on the Port_Ci input signal.
PORT_CI_ICU_RISING_EDGE	An appropriate action shall be executed when a rising edge occurs on the Port_Ci input signal.
PORT_CI_ICU_BOTH_EDGES	An appropriate action shall be executed when a rising edge or falling edge occurs on the Port_Ci input signal.

Definition at line 108 of file Port_Ci_Icu_Ip_Types.h.

6.4.4.2 Port_Ci_Icu_Ip_StatusType

```
enum Port_Ci_Icu_Ip_StatusType
```

Generic error codes.

Enumerator

PORT_CI_IP_STATUS_SUCCESS	Generic operation success status.
PORT_CI_IP_STATUS_ERROR	Generic operation failure status.

Definition at line 123 of file Port_Ci_Icu_Ip_Types.h.

6.4.5 Function Reference

6.4.5.1 Port_Ci_Icu_Ip_EnableNotification()

```
void Port_Ci_Icu_Ip_EnableNotification (
             uint8 instance,
             uint8 hwChannel )
```

Driver function Enable Notification for timestamp.

6.4.5.2 Port_Ci_Icu_Ip_DisableNotification()

```
void Port_Ci_Icu_Ip_DisableNotification (
             uint8 instance,
             uint8 hwChannel )
```

Driver function Disable Notification for timestamp.

6.5 CMP IPL

6.5.1 Detailed Description

CMP HW module.

The low power comparator (LPCMP) module provides a circuit for comparing two analog input voltages. It comprises a comparator (CMP), a DAC and an analog mux (ANMUX). The CMP circuit is designed to operate across the full range of the supply voltage, known as rail-to-rail operation. The DAC is a 256-tap resistor ladder network that provides a selectable voltage reference for applications requiring a voltage reference. The 256-tap resistor ladder network divides the supply reference Vin into 256 voltage levels. A 8-bit digital signal input selects the output voltage level, which varies from Vin to Vin/256. Vin can be selected from two voltage sources, vrefh0 and vrefh1. See the chip-specific LPCMP information for the source of vrefh0 and vrefh1.

The Analog MUX (ANMUX) provides a circuit for selecting an analog input signal from eight channels. One channel is provided by the DAC. Refer to the chip-specific LPCMP information section for details on which device resources are connected to other channels. The mux circuit is designed to operate across the full range of the supply voltage.

Data Structures

- struct Cmp_Ip_StateType

  CMP IP state structure. More...

Enum Reference

- enum Cmp_Ip_StateEType

  CMP initialization status.

Function Reference

- Cmp_Ip_StatusType Cmp_Ip_Init (uint8 instance, const Cmp_Ip_ConfigType ∗userConfig)

  Configure all comparator features with the given configuration structure.
- Cmp_Ip_StatusType Cmp_Ip_Deinit (uint8 instance)

  Reset all register used by ICU driver on current CMP instance.

- void Cmp_Ip_EnableNotification (uint8 instance)

  Enable CMP notification.
- void Cmp_Ip_DisableNotification (uint8 instance)

  Disable CMP notification.
- void Cmp_Ip_SetInterruptActivation (uint8 instance, Cmp_Ip_OutputInterruptTriggerType Edge)

  Set the type of activation for interrupt.
- void Cmp_Ip_EnableInterrupt (uint8 instance)

  Enable CMP interrupt.
- void Cmp_Ip_DisableInterrupt (uint8 instance)

  Disable CMP interrupt.

6.5.2 Data Structure Documentation

6.5.2.1 struct Cmp_Ip_StateType

CMP IP state structure.

This structure is used by the IPL driver for internal logic. The content is populated at initialization time.

Definition at line 218 of file Cmp_Ip_Types.h.

6.5.3 Enum Reference

6.5.3.1 Cmp_Ip_StateEType

```
enum Cmp_Ip_StateEType
```

CMP initialization status.

Definition at line 200 of file Cmp_Ip_Types.h.

6.5.4 Function Reference

6.5.4.1 Cmp_Ip_Init()

```
Cmp_Ip_StatusType Cmp_Ip_Init (
            uint8 instance,
            const Cmp_Ip_ConfigType * userConfig )
```

Configure all comparator features with the given configuration structure.

This function configures the comparator module with the options provided in the config structure.

Parameters

instance	- instance number
config	- the configuration structure

Returns

- CMP_IP_STATUS_SUCCESS : Completed successfully.
- CMP_IP_STATUS_ERROR : Error occurred.

6.5.4.2 Cmp_Ip_Deinit()

```
Cmp_Ip_StatusType Cmp_Ip_Deinit (
            uint8 instance )
```

Reset all register used by ICU driver on current CMP instance.

Parameters

Instance	- The number instance to be deinitialize.

Returns

Cmp_Ip_StatusType

6.5.4.3 Cmp_Ip_EnableNotification()

```
void Cmp_Ip_EnableNotification (
            uint8 instance )
```

Enable CMP notification.

Parameters

Instance	The number of instance for which the notifcation will be enabled.

6.5.4.4 Cmp_Ip_DisableNotification()

```
void Cmp_Ip_DisableNotification (
            uint8 instance )
```

Disable CMP notification.

Parameters

Instance	The number of instance for which the notifcation will be enabled.

6.5.4.5 Cmp_Ip_SetInterruptActivation()

```
void Cmp_Ip_SetInterruptActivation (
            uint8 instance,
            Cmp_Ip_OutputInterruptTriggerType Edge )
```

Set the type of activation for interrupt.

Parameters

Instance	The number of instance for which the interrupt activation will be set.
Edge	

6.5.4.6 Cmp_Ip_EnableInterrupt()

```
void Cmp_Ip_EnableInterrupt (
            uint8 instance )
```

Enable CMP interrupt.

Parameters

Instance	The number of instance for which the interrupt will be enabled.

6.5.4.7 Cmp_Ip_DisableInterrupt()

```
void Cmp_Ip_DisableInterrupt (
            uint8 instance )
```

Disable CMP interrupt.

Parameters

Instance	The number of instance for which the interrupt will be disable.

6.6 FTM

6.6.1 Detailed Description

Macros

- #define ICU_STOP_SEC_CODE

  Ftm_Icu_SetUserAccessAllowed.

6.6.2 Macro Definition Documentation

6.6.2.1 ICU_STOP_SEC_CODE

```
#define ICU_STOP_SEC_CODE
```

Ftm_Icu_SetUserAccessAllowed.

This function is called externally by OS Application

Parameters

in	*FtmBaseAddr*	- The base address of Ftm module.

Definition at line 103 of file Ftm_Icu_Ip_TrustedFunctions.h.

6.7 Icu Driver

6.7.1 Detailed Description

Data Structures

- struct Icu_ChannelConfigType

  Structure that contains ICU channel configuration. More...
- struct Icu_ConfigType

  This type contains initialization data. More...

Types Reference

- typedef uint8 Icu_ChannelStateType

  ICU Channel state type.
- typedef uint16 Icu_ChannelType

  This gives the numeric ID (hardware channel number) of an ICU channel.
- typedef Icu_TimerRegisterWidthType Icu_ValueType

  Type for saving the timer register width value.
- typedef uint16 Icu_MeasurementSubModeType

  Type for saving the ICU measurement submode type.
- typedef void(∗ Icu_NotifyType) (void)

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

Enum Reference

- enum Icu_ModeType

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

  Definition of the measurement mode type.
- enum Icu_ActivationType

  Definition of the type of activation of an ICU channel.

Function Reference

- void Icu_Init (const Icu_ConfigType ∗ConfigPtr)

  This function initializes the driver.
- void Icu_SetActivationCondition (Icu_ChannelType Channel, Icu_ActivationType Activation)

  This function sets the activation-edge for the given channel.
- void Icu_DisableNotification (Icu_ChannelType Channel)

  This function disables the notification of a channel.
- void Icu_EnableNotification (Icu_ChannelType Channel)

  This function enables the notification on the given channel.
- void Icu_ReportWakeupAndOverflow (uint16 Channel, boolean bOverflow)

  This function reports the wakeup and overflow events, if available.
- void Icu_ReportEvents (uint16 Channel, boolean bOverflow)

  This function reports the wakeup event, overflow event and notification, if available.
- void Icu_LogicChStateCallback (uint16 logicChannel, uint8 mask, boolean set)

  Signature of change logic channel state callback function.

Variables

- const Icu_ConfigType ∗ Icu_pCfgPtr [ICU_MAX_PARTITIONS]

  Pointer initialized during init with the address of the received configuration structure.
- Icu_ModeType Icu_CurrentMode

  Saves the current Icu mode.
- volatile Icu_ChannelStateType Icu_aChannelState [ICU_MAX_CHANNEL]

  Stores actual state and configuration of ICU Channels.

6.7.2 Data Structure Documentation

6.7.2.1 struct Icu_ChannelConfigType

Structure that contains ICU channel configuration.

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

Definition at line 538 of file Icu.h.

Data Fields

- boolean Icu_WakeupCapabile

  Channel wakeup capability enable.
- Icu_ActivationType Icu_ActivEdge

  RISING_EDGE, FALLING_EDGE or BOTH_EDGES for EDGE_COUNTER.
- Icu_MeasurementModeType Icu_ChannelMode

  EDGE_DETECT, TIME_STAMP, SIGNAL_MEASUREMENT or EDGE_COUNTER.
- Icu_MeasurementSubModeType Icu_ChannelProperty

  CIRCULAR_BUFFER or LINEAR_BUFFER for TIME_STAMP, DUTY_CYCLE, HIGH_TIME, LOW_TIME or PERIOD_TIME for SIGNAL_MEASUREMENT and RISING_EDGE, FALLING_EDGE or BOTH_EDGES for EDGE_COUNTER.
- Icu_NotifyType Icu_ChannelNotification

  Icu Channel Notification function for TIME_STAMP or EDGE_COUNTER mode.
- const Icu_Ipw_ChannelConfigType ∗ Icu_IpwChannelConfigPtr

  Pointer to the ipw channel pointer configuration.

6.7.2.1.1 Field Documentation

6.7.2.1.1.1 Icu_WakeupCapabile `boolean Icu_WakeupCapabile`

Channel wakeup capability enable.

Definition at line 541 of file Icu.h.

6.7.2.1.1.2 Icu_ActivEdge `Icu_ActivationType Icu_ActivEdge`

RISING_EDGE, FALLING_EDGE or BOTH_EDGES for EDGE_COUNTER.

Definition at line 543 of file Icu.h.

6.7.2.1.1.3 Icu_ChannelMode `Icu_MeasurementModeType Icu_ChannelMode`

EDGE_DETECT, TIME_STAMP, SIGNAL_MEASUREMENT or EDGE_COUNTER.

Definition at line 545 of file Icu.h.

6.7.2.1.1.4 Icu_ChannelProperty `Icu_MeasurementSubModeType Icu_ChannelProperty`

CIRCULAR_BUFFER or LINEAR_BUFFER for TIME_STAMP, DUTY_CYCLE, HIGH_TIME, LOW_TIME or PERIOD_TIME for SIGNAL_MEASUREMENT and RISING_EDGE, FALLING_EDGE or BOTH_EDGES for EDGE_COUNTER.

Definition at line 549 of file Icu.h.

6.7.2.1.1.5 Icu_ChannelNotification `Icu_NotifyType Icu_ChannelNotification`

Icu Channel Notification function for TIME_STAMP or EDGE_COUNTER mode.

Definition at line 551 of file Icu.h.

6.7.2.1.1.6 Icu_IpwChannelConfigPtr `const Icu_Ipw_ChannelConfigType* Icu_IpwChannelConfigPtr`

Pointer to the ipw channel pointer configuration.

Definition at line 564 of file Icu.h.

6.7.2.2 struct Icu_ConfigType

This type contains initialization data.

he 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)

Definition at line 578 of file Icu.h.

Data Fields

- uint8 nNumChannels

  The number of configured logical channels.
- const Icu_ChannelConfigType(∗ Icu_ChannelConfigPtr)[ ]

  Pointer to the list of Icu configured channels.
- uint8 nNumInstances

  The number of IP instances configured.
- const Icu_Ipw_IpConfigType(∗ Icu_IpConfigPtr)[ ]

  Pointer to the list of Icu configured channels.
- const uint8(∗ Icu_IndexChannelMap)[ ]

  channel index in each partition map table
- uint8 u32CoreId

  Core index.

6.7.2.2.1 Field Documentation

6.7.2.2.1.1 nNumChannels `uint8 nNumChannels`

The number of configured logical channels.

Definition at line 581 of file Icu.h.

6.7.2.2.1.2 Icu_ChannelConfigPtr `const Icu_ChannelConfigType(* Icu_ChannelConfigPtr)[]`

Pointer to the list of Icu configured channels.

Definition at line 584 of file Icu.h.

6.7.2.2.1.3 nNumInstances `uint8 nNumInstances`

The number of IP instances configured.

Definition at line 587 of file Icu.h.

6.7.2.2.1.4 Icu_IpConfigPtr `const Icu_Ipw_IpConfigType(* Icu_IpConfigPtr)[]`

Pointer to the list of Icu configured channels.

Definition at line 590 of file Icu.h.

6.7.2.2.1.5 Icu_IndexChannelMap `const uint8(* Icu_IndexChannelMap)[]`

channel index in each partition map table

Definition at line 593 of file Icu.h.

6.7.2.2.1.6 u32CoreId `uint8 u32CoreId`

Core index.

Definition at line 596 of file Icu.h.

6.7.3 Types Reference

6.7.3.1 Icu_ChannelStateType

```
typedef uint8 Icu_ChannelStateType
```

ICU Channel state type.

Definition at line 221 of file Icu_Types.h.

6.7.3.2 Icu_ChannelType

```
typedef uint16 Icu_ChannelType
```

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

Definition at line 226 of file Icu_Types.h.

6.7.3.3 Icu_ValueType

```
typedef Icu_TimerRegisterWidthType Icu_ValueType
```

Type for saving the timer register width value.

Definition at line 231 of file Icu_Types.h.

6.7.3.4 Icu_MeasurementSubModeType

```
typedef uint16 Icu_MeasurementSubModeType
```

Type for saving the ICU measurement submode type.

Definition at line 259 of file Icu_Types.h.

6.7.3.5 Icu_NotifyType

```
typedef void(* Icu_NotifyType) (void)
```

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

Definition at line 264 of file Icu_Types.h.

6.7.4 Enum Reference

6.7.4.1 Icu_ModeType

```
enum Icu_ModeType
```

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

Enumerator

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

Definition at line 103 of file Icu_Types.h.

6.7.4.2 Icu_MeasurementModeType

```
enum Icu_MeasurementModeType
```

Definition of the measurement mode type.

Enumerator

ICU_MODE_SIGNAL_EDGE_DETECT	Mode for detecting edges.
ICU_MODE_SIGNAL_MEASUREMENT	Mode for measuring different times between various configurable edges.
ICU_MODE_TIMESTAMP	Mode for capturing timer values on configurable edges.
ICU_MODE_EDGE_COUNTER	Mode for counting edges on configurable edges.

Definition at line 129 of file Icu_Types.h.

6.7.4.3 Icu_ActivationType

```
enum Icu_ActivationType
```

Definition of the type of activation of an ICU channel.

Enumerator

ICU_RISING_EDGE	An appropriate action shall be executed when a rising edge occurs on the ICU input signal.
ICU_FALLING_EDGE	An appropriate action shall be executed when a falling edge occurs on the ICU input signal.
ICU_BOTH_EDGES	An appropriate action shall be executed when either a rising or falling edge occur on the ICU input signal.

Definition at line 176 of file Icu_Types.h.

6.7.5 Function Reference

6.7.5.1 Icu_Init()

```
void Icu_Init (
            const Icu_ConfigType * ConfigPtr )
```

This function initializes the driver.

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 Config↩ Ptr. 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).

Parameters

in	*ConfigPtr*	Pointer to a selected configuration structure.

Returns

void

6.7.5.2 Icu_SetActivationCondition()

```
void Icu_SetActivationCondition (
            Icu_ChannelType Channel,
            Icu_ActivationType Activation )
```

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

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

Parameters

in	*Channel*	Logical number of the ICU channel
in	*Activation*	Type of activation.

Returns

void

Precondition

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

6.7.5.3 Icu_DisableNotification()

```
void Icu_DisableNotification (
            Icu_ChannelType Channel )
```

This function disables the notification of a channel.

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

Parameters

in	*Channel*	Logical number of the ICU channel

Returns

void

Precondition

Icu_Init must be called before.

6.7.5.4 Icu_EnableNotification()

```
void Icu_EnableNotification (
            Icu_ChannelType Channel )
```

This function enables the notification on the given channel.

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

Parameters

in	*Channel*	Logical number of the ICU channel

Returns

void

Precondition

Icu_Init must be called before.

6.7.5.5 Icu_ReportWakeupAndOverflow()

```
void Icu_ReportWakeupAndOverflow (
            uint16 Channel,
            boolean bOverflow )
```

This function reports the wakeup and overflow events, if available.

This function reports the wakeup and overflow events, if available. Called from hardware interrupt routine and route to user overflow handler

Parameters

in	*Channel*	Hardware number identifier of the ICU channel
in	*bOverflow*	Parameter that indicates the source of report is an overflow

Returns

void

Precondition

Icu_Init must be called before.

6.7.5.6 Icu_ReportEvents()

```
void Icu_ReportEvents (
            uint16 Channel,
            boolean bOverflow )
```

This function reports the wakeup event, overflow event and notification, if available.

This function reports the wakeup event, overflow event and notification, if available

Parameters

in	*Channel*	Harware number identifier of the ICU channel
in	*overflow*	Parameter that indicates the source of report is an overflow

Returns

void

Precondition

Icu_Init must be called before.

6.7.5.7 Icu_LogicChStateCallback()

```
void Icu_LogicChStateCallback (
            uint16 logicChannel,
            uint8 mask,
            boolean set )
```

Signature of change logic channel state callback function.

Parameters

logicChannel	Logical number of the ICU channel
mask	Bit mark
set	Set value

6.7.6 Variable Documentation

6.7.6.1 Icu_pCfgPtr

```
const Icu_ConfigType* Icu_pCfgPtr[ICU_MAX_PARTITIONS]  [extern]
```

Pointer initialized during init with the address of the received configuration structure.

Will be used by all functions to access the configuration data.

6.7.6.2 Icu_CurrentMode

```
Icu_ModeType Icu_CurrentMode  [extern]
```

Saves the current Icu mode.

6.7.6.3 Icu_aChannelState

```
volatile Icu_ChannelStateType Icu_aChannelState[ICU_MAX_CHANNEL]  [extern]
```

Stores actual state and configuration of ICU Channels.

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.

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

© 2022 NXP B.V.