User Manual

for MPC5634M ICU Driver

Document Number: UM14ICUASR3.0R2.0.0
Rev. 2.0

Contents

Section Number **Title** **Page**

Chapter 1
Revision History

Chapter 2
Introduction

Chapter 3
Driver

Chapter 1
Revision History

Table 1-1. Revision History

Revision	Date	Author	Description
1.0	27.01.2011	Chandrakanth.P	Initial Version
2.0	27.11.2011	Syed Hussaini	Updated for Monacco RTM2.0.0

Chapter 2
Introduction

This User Manual describes Freescale Semiconductor AUTOSAR microcontroller abstraction layer (MCAL) Input Capture Unit (ICU) for MPC5634M microcontroller.

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 Freescale Semiconductor:

Table 2-1. MPC5634M Supported Derivatives

Freescale Semiconductor	mpc5634m_bga208, mpc5634m_qfp144, mpc5634m_qfp176

All of the above microcontroller devices are collectively named as **MPC5634M**.

2.2 Overview

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

AUTOSAR

- paves the way for innovative electronic systems that further improve performance, safety and environmental friendliness.

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

2.3 About this Manual

This Technical Reference employs the following typographical conventions:

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

Italic font: Italic typeface is used for code snippets in the text. Note that C language modifiers such "const" or "volatile" are sometimes omitted to improve readability of the presented code.

2.4 Acronyms and Definitions

Table 2-2. Acronyms and Definitions

Abbreviation and Definitions	Description
BSW	Basic Software
DEM	Diagnostic Event Manager
DET	Development Error Tracer
ECU	Electronic Control Unit
ICU	Input Capture Unit
ISR	interrupt Service Routine
OS	Operating System
RAM	Random Access Memory
ROM	Read-only Memory
MCU	Microcontroller Unit
GUI	Graphical User Interface
EcuM	ECU state Manager
API	Application Programming Interface
PB Variant	Post Build Variant
PC Variant	Pre Compile Variant

2.5 Reference List

Table 2-3. Reference List

#	Title	Version
1	AUTOSAR 3.0ICU Driver Software Specification Document.	V2.2.0 R3.0 Rev 0001
2	MPC5634M Reference Manual	Rev. 6, 4 October 2011

Chapter 3
Driver

3.1 Requirements

Requirements for this driver are detailed in the AUTOSAR 3.0ICUDriver Software Specification document (See Table **Reference List**).

3.2 Driver Design Sumary

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

- Period time measurement
- Edge detection and notification
- Edge counting (with or without hardware gating)
- Edge time stamping

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

For signal measuring a capture timer and at least one capture register is needed.

3.3 Hardware channel availability

The eMIOS module of MPC5634M supports all above mentioned features of ICU module.

Table 3-1. ICU Hardware channels availability for MPC5634Mfamily

Device	Total eMIOS channels	Total External interrupt channels
MPC5634M	24 ch, 16-bit	16 ch

3.4 Additional Requirements and Deviations

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

Table 3-2. Deviations Status Column Description

Term	Definition
N/A	Not available
N/T	Not testable
N/S	Out of scope
N/I	Not implemented
N/F	Not fully implemented

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

Table 3-3. ICU Deviations Table

Requirement	Status	Description	Notes
ICU003	N/I	Production errors shall be reported to the Diagnostic Event Manager (DEM).	No Production Error

Table continues on the next page...

Table 3-3. ICU Deviations Table (continued)

Requirement	Status	Description	Notes
ICU039	N/F	Configuration of Signal Measurement - This container contains the configuration (parameters) in case the measurement mode is “ICU_MODE_SIGNAL_MEASUREMENT” The definition for each Channel shall contain: Common parameters § Wakeup capability (true / false) § Default Start Edge § Hardware Specific Settings per channel § Measurement Mode - Signal Edge Detection / Notification - Signal Measurement - Timestamp - Edge Counter Specific parameters § If measurement mode is “signal edge detection” the notification function for signal notification shall be configurable § If measurement mode is “signal measurement” the property that could be measured shall be configurable. The values shall be: - High Time - Low Time - Period Time - Duty Cycle Values (High Time and Period Time) § If measurement mode is “timestamp measurement”, buffer handling shall be configurable. The values shall be: - Circular buffer handling - Linear buffer handling Also the notification function for notifying the number of requested timestamps shall be configurable § If measurement mode is “edge counter”, the counting mode (activation edge) shall be configurable. The values shall be: - Rising Edge - Falling Edge - Both edges § If the channel is configured as wakeup capable, - the callout function for validation of wakeup reason shall be configurable - The value transmitted to the EcuM shall be configurable § Assigned capture register(s) (can also be none for channels which provide only edge detection like an external interrupt) Assigned capture timer (can also be none for channels which provide only edge detection like an external interrupt)	For memory space saving capture register and capture timer were omitted, indexed access is used. Capture registers and capture timer are not enough for this architecture, it would require also the control register and the status register. Because registers in memory are very well organized, the access to these registers was done by macros that access memory with offset depending the channel used.
ICU052	N/S	Icu_Init - If the register can affect several hardware modules and if it is an I/O register it shall be initialized by the PORT driver.	Not an ICU driver requirement
ICU053	N/S	Icu_Init - If the register can affect several hardware modules and if it is not an I/O register it shall be initialized by the MCU driver.	Not an ICU driver requirement
ICU113	N/I	The detection of production code errors cannot be switched off. Specification of ICU Driver	No Production Error

Table continues on the next page...

Table 3-3. ICU Deviations Table (continued)

Requirement	Status	Description	Notes
ICU116	N/I	The module shall include the Dem.h file. By this inclusion the API's to reporterrors as well as the required Event Id symbols are included.This specification defines the name of the Event Id symbols which are provided byXML to the DEM configuration tool. The DEM configuration tool assigns ECUdependent values to the Event Id symbols and publishes the symbols inDem_IntErrId.h.	No Production Error
ICU117	N/I	Values for production code event ID's are assigned externally by the configuration of the DEM. They are published in the file Dem_IntErrId.h and included via Dem.h.	No Production Error
ICU128	N/S	One-time writable registers that require initialization directly after reset shall be initialized by the startup code.	Not an ICU driver requirement
ICU129	N/S	All other registers shall be initialized by the startup code.	Not an ICU driver requirement
ICU131	N/I	The ICU driver shall describe which other modules (in which versions) are required.This description shall be done by the implementer.	Port Driver Mcu Driver (FMPLL)

3.5 Function Definitions

APIs of all functions supported by the driver are as per AUTOSAR ICU Driver software specification Version 3.0.

3.5.1 Function Icu_Init

This function initializes the driver.

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

Table 3-4. Icu_Init Arguments

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

This service is a non reentrant function used for driver initialization. The Initialization function shall initialize all relevant registers of the configured hardware with the values of the structure referenced by the parameter ConfigPtr. If the hardware allows for only

one usage of the register, the driver module implementing that functionality is responsible for initializing the register. The initialization function of this module shall always have a pointer as a parameter, even though for Variant PC no configuration set shall be given. Instead a NULL pointer shall be passed to the initialization function. The Icu module environment shall not call Icu_Init during a running operation (e. g. timestamp measurement or edge counting).

Remarks:
- Covers ICU002, ICU111, ICU220, ICU023, ICU054, ICU060
- Implements DICU20100, DICU01015, DICU10001

3.5.2 Function Icu_DeInit

This function de-initializes the ICU module.

Prototype: `void Icu_DeInit(void);`

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

pre:: Icu_Init must be called before.

Remarks:
- Covers ICU002, ICU111, ICU112, ICU022, ICU048, ICU091, ICU036
- Implements DICU01002, DICU10002

3.5.3 Function Icu_SetMode

This function sets the ICU mode.

Prototype: `void Icu_SetMode(Icu_ModeType Mode);`

Table 3-5. Icu_SetMode Arguments

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

This service is a non reentrant function used for ICU mode selection. This service shall set the operation mode to the given mode parameter. This service can be called during running operations. If so, an ongoing operation that generates interrupts on a wakeup

capable channel like e.g. time stamping or edge counting might lead to the ICU module not being able to properly enter sleep mode. This is then a system or ECU configuration issue not a problem of this specification.

pre:: Icu_Init must be called before.

Remarks:
- Covers ICU002, ICU111, ICU022, ICU048, ICU095, ICU125, ICU011, ICU012
- Implements DICU01020, DICU10003

3.5.4 Function Icu_SetClockMode

This function changes the channel prescaler.

Prototype: `void Icu_SetClockMode(Icu_SelectPrescalerType Prescaler);`

Table 3-6. Icu_SetClockMode Arguments

Type	Name	Direction	Description
`Icu_SelectPrescalerType`	Prescaler	**input**	- Prescaler type (Normal or Alternate)

This function sets all channels prescalers based on the input mode.

pre:: Icu_Init must be called before.

3.5.5 Function Icu_DisableWakeup

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

Prototype: `void Icu_DisableWakeup(Icu_ChannelType Channel);`

Table 3-7. Icu_DisableWakeup Arguments

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

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

pre:: Icu_Init must be called before.

Remarks:

- Covers ICU002, ICU111, ICU022, ICU048, ICU024, ICU059
- Implements DICU01005, DICU10004

3.5.6 Function Icu_EnableWakeup

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

Prototype: `void Icu_EnableWakeup(Icu_ChannelType Channel);`

Table 3-8. Icu_EnableWakeup Arguments

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

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

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

Remarks:
- Covers ICU002, ICU111, ICU022, ICU048, ICU155, ICU156
- Implements DICU01008, DICU10005

3.5.7 Function Icu_SetActivationCondition

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

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

Table 3-9. Icu_SetActivationCondition Arguments

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

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

pre:: Icu_Init must be called before. The channel must be properly configured ICU_MODE_SIGNAL_EDGE_DETECT,ICU_MODE_TIMESTAMP, ICU_MODE_EDGE_COUNTER.

Remarks:
- Covers ICU002, ICU111, ICU112, ICU022, ICU048, ICU159, ICU043
- Implements DICU01019, DICU10006

3.5.8 Function Icu_DisableNotification

This function disables the notification of a channel.

Prototype: `void Icu_DisableNotification(Icu_ChannelType Channel);`

Table 3-10. Icu_DisableNotification Arguments

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

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

pre:: Icu_Init must be called before.

Remarks:
- Covers ICU002, ICU111, ICU112, ICU022, ICU048, ICU160
- Implements DICU01004, DICU10007

3.5.9 Function Icu_EnableNotification

This function enables the notification on the given channel.

Prototype: `void Icu_EnableNotification(Icu_ChannelType Channel);`

Table 3-11. Icu_EnableNotification Arguments

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

This function is reentrant and enables the notification on the given channel.

pre:: Icu_Init must be called before.

Remarks:
- Covers ICU002, ICU111, ICU112, ICU022, ICU048, ICU161
- Implements DICU01007, DICU10008

3.5.10 Function Icu_GetInputState

This function returns the status of the ICU input.

Prototype: `Icu_InputStateType Icu_GetInputState(Icu_ChannelType Channel);`

Table 3-12. Icu_GetInputState Arguments

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

This service is reentrant shall return the status of the ICU input.

Only channels which are configured for the following Icu_MeasurementMode shall be supported:

ICU_MODE_SIGNAL_EDGE_DETECT,

ICU_MODE_SIGNAL_MEASUREMENT.

Return: Icu_InputStateType - ICU_ACTIVE: An activation edge has been detected,ICU_IDLE: No activation edge has been detected since the last call of `Icu_GetInputState()` or `Icu_Init()`.

pre:: Icu_Init must be called before.

Remarks:
- Covers ICU002, ICU111, ICU112, ICU022, ICU048, ICU162
- Implements DICU01011, DICU10009

3.5.11 Function Icu_StartTimestamp

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

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

Table 3-13. Icu_StartTimestamp Arguments

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

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

pre:: Icu_Init must be called before.

Remarks:
- Covers ICU002, ICU111, ICU112, ICU022, ICU048, ICU163, ICU120, ICU108
- Implements DICU01022, DICU10010

3.5.12 Function Icu_StopTimestamp

This function stops the timestamp measurement of the given channel.

Prototype: `void Icu_StopTimestamp(Icu_ChannelType Channel);`

Table 3-14. Icu_StopTimestamp Arguments

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

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

pre:: Icu_Init must be called before.

Remarks:
- Covers ICU002, ICU111, ICU112, ICU022, ICU048, ICU164, ICU166
- Implements DICU01024, DICU10011

3.5.13 Function Icu_GetTimestampIndex

This function reads the timestamp index of the given channel.

Prototype: `Icu_IndexType Icu_GetTimestampIndex(Icu_ChannelType Channel);`

Table 3-15. Icu_GetTimestampIndex Arguments

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

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

Return: Icu_IndexType - Timestamp index of the given channel

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

Remarks:
- Covers ICU002, ICU111, ICU112, ICU022, ICU048, ICU169, ICU107, ICU004
- Implements DICU01013, DICU10012

3.5.14 Function Icu_ResetEdgeCount

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

Prototype: `void Icu_ResetEdgeCount(Icu_ChannelType Channel);`

Table 3-16. Icu_ResetEdgeCount Arguments

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

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

pre:: Icu_Init must be called before.

Remarks:
- Covers ICU002, ICU111, ICU112, ICU022, ICU048, ICU171
- Implements DICU01018, DICU10013

3.5.15 Function Icu_EnableEdgeCount

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

Prototype: `void Icu_EnableEdgeCount(Icu_ChannelType Channel);`

Table 3-17. Icu_EnableEdgeCount Arguments

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

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

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

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

Remarks:
- Covers ICU002, ICU111, ICU112, ICU022, ICU048, ICU172
- Implements DICU01006, DICU10014

3.5.16 Function Icu_DisableEdgeCount

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

Prototype: `void Icu_DisableEdgeCount(Icu_ChannelType Channel);`

Table 3-18. Icu_DisableEdgeCount Arguments

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

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

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

Remarks:
- Covers ICU002, ICU111, ICU112, ICU022, ICU048, ICU173
- Implements DICU01003, DICU10015

3.5.17 Function Icu_GetEdgeNumbers

This function reads the number of counted edges.

Prototype: `Icu_EdgeNumberType Icu_GetEdgeNumbers(Icu_ChannelType Channel);`

Table 3-19. Icu_GetEdgeNumbers Arguments

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

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

Return: Icu_EdgeNumberType - Number of the counted edges.

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

Remarks:
- Covers ICU002, ICU111, ICU112, ICU022, ICU048, ICU107, ICU174, ICU175, ICU004
- Implements DICU01010, DICU10016

3.5.18 Function Icu_StartSignalMeasurement

This function starts the measurement of signals.

Prototype: `void Icu_StartSignalMeasurement(Icu_ChannelType Channel);`

Table 3-20. Icu_StartSignalMeasurement Arguments

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

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

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

Remarks:
- Covers ICU002, ICU111, ICU112, ICU022, ICU048, ICU176
- Implements DICU01021, DICU10017

3.5.19 Function Icu_StopSignalMeasurement

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

Prototype: `void Icu_StopSignalMeasurement(Icu_ChannelType Channel);`

Table 3-21. Icu_StopSignalMeasurement Arguments

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

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

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

Remarks:
- Covers ICU002, ICU111, ICU112, ICU022, ICU048, ICU177
- Implements DICU01023, DICU10018

3.5.20 Function Icu_GetTimeElapsed

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

Prototype: `Icu_ValueType Icu_GetTimeElapsed(Icu_ChannelType Channel);`

Table 3-22. Icu_GetTimeElapsed Arguments

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

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

Return: Icu_ValueType - The elapsed Signal Low Time for the given channel

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

Remarks:
- Covers ICU002, ICU111, ICU112, ICU022, ICU048, ICU107, ICU178, ICU004, ICU179, ICU136
- Implements DICU01012, DICU10019

3.5.21 Function Icu_GetDutyCycleValues

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

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

Table 3-23. Icu_GetDutyCycleValues Arguments

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

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

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

Remarks:
- Covers ICU002, ICU111, ICU112, ICU022, ICU048, ICU180, ICU107, ICU004, ICu181, ICU137
- Implements DICU01009, DICU10020

3.5.22 Function Icu_GetVersionInfo

This service returns the version information of this module.

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

Table 3-24. Icu_GetVersionInfo Arguments

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

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

The version information includes:

- Module Id
- Vendor Id
- Vendor specific version numbers

If source code for caller and callee of this function is available this function should be realized as a macro. The macro should be defined in the modules header file.

Remarks:
- Covers ICU005
- Implements DICU01014

3.6 Structure Definitions

Data structures supported by the driver are as per AUTOSAR ICU Driver software specification Version 3.0.

3.6.1 Structure Icu_ConfigType

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

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

Declaration

```
typedef struct
                    {
                      const Icu_ChannelConfigType * Icu_ChannelConfigPtr,
                              const Icu_ChannelType * Icu_ChannelId,
                              const Icu_ChannelType * Icu_ChannelIndex,
                              const Icu_ChannelType * Icu_HWMap,
                              const Icu_ChannelType * Icu_HWWKMap,
                              const Icu_ChannelType Icu_MaxChannels
                    } Icu_ConfigType;
```

Table 3-25. Structure Icu_ConfigType member description

Member	Description
Icu_ChannelConfigPtr	Pointer to Icu channel configuration.
Icu_ChannelId	Array for mapping the Icu channels to hardware channels.
Icu_ChannelIndex	This index relates the Icu Channel number with the respective global variable, depending on the measurement mode Each kind of measurement mode has an array(s) in the ICU driver.
Icu_HWMap	This index relates the Hardware channels with the respective ICU channel.
Icu_HWWKMap	This index relates the Wakeup pins with the respective ICU channel.
Icu_MaxChannels	The number of configured channels.

Remarks:

- Covers ICU038, ICU039,ICU190
- Implements DICU50200

3.6.2 Structure Icu_ChannelConfigType

Structure that contains ICU channel configuration. It contains the information like Icu Channel Mode, Channel Notification function, Overflow Notification function.

Declaration

```
typedef struct
                    {
                      Icu_MeasurementModeType Icu_Channel_Mode,
                              Icu_NotifyType Icu_Channel_Notification,
                              Icu_NotifyType Icu_Channel_OverFlowNotification,
                              Icu_MeasurementSubModeType Icu_Channel_Property,
                              Icu_WakeupValue Icu_Channel_WakeupValue,
                              Icu_ParamType Icu_ParamValue
                    } Icu_ChannelConfigType;
```

Table 3-26. Structure Icu_ChannelConfigType member description

Member	Description
Icu_Channel_Mode	Icu Measurement mode type i.e EDGE_DETECT, TIME_STAMP, SIGNAL_MEASUREMENT or EDGE_COUNTER.
Icu_Channel_Notification	Icu Channel Notification function for TIME_STAMP or EDGE_COUNTER mode.
Icu_Channel_OverFlowNotification	Icu Custom notification function. Icu Channel Overflow Notification function
Icu_Channel_Property	Icu Channel Property type i.e 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.

Table continues on the next page...

Table 3-26. Structure Icu_ChannelConfigType member description (continued)

Member	Description
Icu_Channel_WakeupValue	EcuM wakeup source Id.
Icu_ParamValue	Configuration parameters of IP registers.

Remarks:

- Implements DICU50202

3.6.3 Structure Icu_DutyCycleType

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

Declaration

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

Table 3-27. Structure Icu_DutyCycleType member description

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

3.7 Enum Definitions

Data Enums supported by the driver are as per AUTOSAR ICU Driver software specification Version 3.0.

3.7.1 Enumeration Icu_SelectPrescalerType

Enumeration of available prescalers.

Table 3-28. Enumeration Icu_SelectPrescalerType Values

Value	Description
ICU_NORMAL = 0x0U	Default channel prescaler option
ICU_ALTERNATE = 0x1U	Alternative prescaler to be used depending on user application (e.g. low power modes)

3.7.2 Enumeration Icu_ActivationType

Definition of the type of activation of an ICU channel.

Table 3-29. Enumeration Icu_ActivationType Values

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

3.7.3 Enumeration Icu_InputStateType

Input state of an ICU channel.

Table 3-30. Enumeration Icu_InputStateType Values

Value	Description
ICU_ACTIVE = 0U	ICU_ACTIVE = An activation edge has been detected
ICU_IDLE	ICU_IDLE = No activation edge has been detected since the last call of `Icu_GetInputState()` or `Icu_Init()`.

3.7.4 Enumeration Icu_MeasurementModeType

Definition of the measurement mode type.

Table 3-31. Enumeration Icu_MeasurementModeType Values

Value	Description
ICU_MODE_SIGNAL_EDGE_DETECT = 0U	ICU_MODE_SIGNAL_EDGE_DETECT = Mode for detecting edges
ICU_MODE_SIGNAL_MEASUREMENT	ICU_MODE_SIGNAL_MEASUREMENT = Mode for measuring different times between various configurable edges
ICU_MODE_TIMESTAMP	ICU_MODE_TIMESTAMP = Mode for capturing timer values on configurable edges
ICU_MODE_EDGE_COUNTER	ICU_MODE_EDGE_COUNTER = Mode for counting edges on configurable edges

3.7.5 Enumeration Icu_ModeType

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

Table 3-32. Enumeration Icu_ModeType Values

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

3.7.6 Enumeration Icu_SignalMeasurementPropertyType

Definition of the measurement property type.

Table 3-33. Enumeration Icu_SignalMeasurementPropertyType Values

Value	Description
ICU_LOW_TIME = 0U	ICU_LOW_TIME = The channel is configured for reading the elapsed Signal Low Time
ICU_HIGH_TIME	ICU_HIGH_TIME = The channel is configured for reading the elapsed Signal High Time
ICU_PERIOD_TIME	ICU_PERIOD_TIME = The channel is configured for reading the elapsed Signal Period Time
ICU_DUTY_CYCLE	ICU_DUTY_CYCLE = The channel is configured to read values which are needed for calculating the duty cycle (coherent Active and Period Time).

3.7.7 Enumeration Icu_TimestampBufferType

Definition of the timestamp measurement property type.

Table 3-34. Enumeration Icu_TimestampBufferType Values

Value	Description
ICU_LINEAR_BUFFER = 0U	ICU_LINEAR_BUFFER = The buffer will just be filled once
ICU_CIRCULAR_BUFFER	ICU_CIRCULAR_BUFFER = After reaching the end of the buffer, the driver restarts at the beginning of the buffer

3.7.8 Enumeration Icu_WakeupCapableType

Definition of the type of wake up capability of an ICU channel.

Table 3-35. Enumeration Icu_WakeupCapableType Values

Value	Description
ICU_WAKEUP_NOTCAPABLE = 0x0U	ICU_WAKEUP_NOTCAPABLE = Channel is not wakeup capable
ICU_WAKEUP_CAPABLE = 0x1U	ICU_WAKEUP_CAPABLE = Channel is wakeup capable.

3.8 Typedef Definitions

Data Typedefs supported by the driver are as per AUTOSAR ICU Driver software specification Version 3.0.

3.8.1 Typedef Icu_ChannelType

Type Definitions.

Type: uint8

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

3.8.2 Typedef Icu_MeasurementSubModeType

Type:`uint8`

Type for saving the ICU measurement submode type

3.8.3 Typedef Icu_ParamType

Type:`uint32`

Icu_ParamType is defined as a uint32. The Icu_ParamValue contains combined bit fields for initialization options, for different registers.

3.9 Define Definitions

3.9.1 Define ICU_DEINIT_ID

API service ID for Icu_DeInit function.

Parameters used when raising an error/exception

Definition:`#define ICU_DEINIT_ID 0x01U`

3.9.2 Define ICU_DISABLEEDGECOUNT_ID

API service ID for Icu_DisableEdgeCount function.

Parameters used when raising an error/exception

Definition:`#define ICU_DISABLEEDGECOUNT_ID 0x0EU`

3.9.3 Define ICU_DISABLENOTIFICATION_ID

API service ID for Icu_DisableNotification function.

Parameters used when raising an error/exception

Definition:`#define ICU_DISABLENOTIFICATION_ID 0x06U`

3.9.4 Define ICU_DISABLEWAKEUP_ID

API service ID for Icu_DisableWakeup function.

Parameters used when raising an error/exception

Definition:`#define ICU_DISABLEWAKEUP_ID 0x03U`

3.9.5 Define ICU_ENABLEEDGECOUNT_ID

API service ID for Icu_EnableEdgeCount function.

Parameters used when raising an error/exception

Definition:`#define ICU_ENABLEEDGECOUNT_ID 0x0DU`

3.9.6 Define ICU_ENABLENOTIFICATION_ID

API service ID for Icu_EnableNotification function.

Parameters used when raising an error/exception

Definition: `#define ICU_ENABLENOTIFICATION_ID 0x07U`

3.9.7 Define ICU_ENABLEWAKEUP_ID

API service ID for Icu_EnableWakeup function.

Parameters used when raising an error/exception

Definition: `#define ICU_ENABLEWAKEUP_ID 0x04U`

3.9.8 Define ICU_GETDUTYCYCLEVALUES_ID

API service ID for Icu_GetDutyCycleValues function.

Parameters used when raising an error/exception

Definition: `#define ICU_GETDUTYCYCLEVALUES_ID 0x11U`

3.9.9 Define ICU_GETEDGENUMBERS_ID

API service ID for Icu_GetEdgeNumbers function.

Parameters used when raising an error/exception

Definition: `#define ICU_GETEDGENUMBERS_ID 0x0FU`

3.9.10 Define ICU_GETINPUTSTATE_ID

API service ID for Icu_GetInputState function.

Parameters used when raising an error/exception

Definition: `#define ICU_GETINPUTSTATE_ID 0x08U`

3.9.11 Define ICU_GETTIMEELAPSED_ID

API service ID for Icu_GetTimeElapsed function.

Parameters used when raising an error/exception

Definition: `#define ICU_GETTIMEELAPSED_ID 0x10U`

3.9.12 Define ICU_GETTIMESTAMPINDEX_ID

API service ID for Icu_GetTimestampIndex function.

Parameters used when raising an error/exception

Definition: `#define ICU_GETTIMESTAMPINDEX_ID 0x0BU`

3.9.13 Define ICU_GETVERSIONINFO_ID

API service ID for Icu_GetVersionInfo function.

Parameters used when raising an error/exception

Definition: `#define ICU_GETVERSIONINFO_ID 0x12U`

3.9.14 Define ICU_INIT_ID

API service ID for Icu_Init function.

Parameters used when raising an error/exception

Definition: `#define ICU_INIT_ID 0x00U`

3.9.15 Define ICU_RESETEDGECOUNT_ID

API service ID for Icu_ResetEdgeCount function.

Parameters used when raising an error/exception

Definition: `#define ICU_RESETEDGECOUNT_ID 0x0CU`

3.9.16 Define ICU_SET_CLOCK_MODE_ID

API service ID for Icu_SetClockMode function.

Parameters used when raising an error/exception

Definition: `#define ICU_SET_CLOCK_MODE_ID 0x7BU`

3.9.17 Define ICU_SETACTIVATIONCONDITION_ID

API service ID for Icu_SetActivationCondition function.

Parameters used when raising an error/exception

Definition: `#define ICU_SETACTIVATIONCONDITION_ID 0x05U`

3.9.18 Define ICU_SETMODE_ID

API service ID for Icu_SetMode function.

Parameters used when raising an error/exception

Definition: `#define ICU_SETMODE_ID 0x02U`

3.9.19 Define ICU_STARTSIGNALMEASUREMENT_ID

API service ID for Icu_StartSignalMeasurement function.

Parameters used when raising an error/exception

Definition: `#define ICU_STARTSIGNALMEASUREMENT_ID 0x13U`

3.9.20 Define ICU_STARTTIMESTAMP_ID

API service ID for Icu_StartTimestamp function.

Parameters used when raising an error/exception

Definition: `#define ICU_STARTTIMESTAMP_ID 0x09U`

3.9.21 Define ICU_STOPSIGNALMEASUREMENT_ID

API service ID for Icu_StopSignalMeasurement function.

Parameters used when raising an error/exception

Definition: `#define ICU_STOPSIGNALMEASUREMENT_ID 0x14U`

3.9.22 Define ICU_STOPTIMESTAMP_ID

API service ID for Icu_StopTimestamp function.

Parameters used when raising an error/exception

Definition:`#define ICU_STOPTIMESTAMP_ID 0x0AU`

3.9.23 Define ICU_E_ALREADY_INITIALIZED

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

Definition:`#define ICU_E_ALREADY_INITIALIZED 0x17U`

3.9.24 Define ICU_E_BUSY_OPERATION

API service Icu_SetMode is called while a running operation.

Definition:`#define ICU_E_BUSY_OPERATION 0x16U`

3.9.25 Define ICU_E_EDGECOUNTING_OVERFLOW

API Icu_GetEdgeNumbers service called when the Counter rolls over.

Definition:`#define ICU_E_EDGECOUNTING_OVERFLOW 0x18U`

3.9.26 Define ICU_E_MEASUREMENT_OVERFLOW

API Icu_GetTimeElapsed service called when the Time elapsed overflows.

Definition:`#define ICU_E_MEASUREMENT_OVERFLOW 0x1AU`

3.9.27 Define ICU_E_NOT_STARTED

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

Definition:`#define ICU_E_NOT_STARTED 0x15U`

3.9.28 Define ICU_E_PARAM_ACTIVATION

API service used with an invalid or not feasible activation.

Definition: `#define ICU_E_PARAM_ACTIVATION 0x0CU`

3.9.29 Define ICU_E_PARAM_BUFFER_PTR

API service used with an invalid application-buffer pointer.

Definition: `#define ICU_E_PARAM_BUFFER_PTR 0x0DU`

3.9.30 Define ICU_E_PARAM_BUFFER_SIZE

API service used with an invalid buffer size.

Definition: `#define ICU_E_PARAM_BUFFER_SIZE 0x0EU`

3.9.31 Define ICU_E_PARAM_CHANNEL

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

Definition: `#define ICU_E_PARAM_CHANNEL 0x0BU`

3.9.32 Define ICU_E_PARAM_CLOCK_MODE

API Icu_SetClockMode service called with wrong parameter.

Definition: `#define ICU_E_PARAM_CLOCK_MODE 0x7AU`

3.9.33 Define ICU_E_PARAM_CONFIG

API Icu_Init service called with wrong parameter.

Definition: `#define ICU_E_PARAM_CONFIG 0x0AU`

3.9.34 Define ICU_E_PARAM_MODE

API service Icu_SetMode used with an invalid mode.

Definition:`#define ICU_E_PARAM_MODE 0x0FU`

3.9.35 Define ICU_E_TIMESTAMP_OVERFLOW

API Icu_GetTimestampIndex service called when the Time stamp count overflows.

Definition:`#define ICU_E_TIMESTAMP_OVERFLOW 0x19U`

3.9.36 Define ICU_E_UNINIT

API service used without module initialization.

Definition:`#define ICU_E_UNINIT 0x14U`

3.10 Symbolic Names DISCLAIMER

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

#define <Container_Short_Name> <Container_ID>

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

3.11 Configuration Parameters

As per the AUTOSAR specification the driver has two types of configurations parameters:**Pre-Compile** parameters and **Post-Build** parameters.

These can be selected using “IMPLEMENTATION_CONFIG_VARIANT”

Pre-Compile parameters are stored in the file "Icu_Cfg.h" and "Icu_Cfg.c". Post-Build parameters are stored in the file "Icu_PBcfg.c".

The files to be used for different configuration types are listed below:

1. **Variant PC:** Icu_Cfg.h, Icu_Cfg.c
2. **Variant PB**: Icu_Cfg.h, Icu_PBcfg.c
3. **Variant LT**: Not Applicable

A section for **Icu_PBcfg.c** file is needed in linker file to place the post build configuration in desired location. Please refer to section-8 of AUTOSAR_SWS_C_ImplementationRules file for complete details on configuration types.

3.11.1 Plugin interface

The following picture gives an example of configuration screen for all ICU parameters in Tresos® Studio configuration tool.

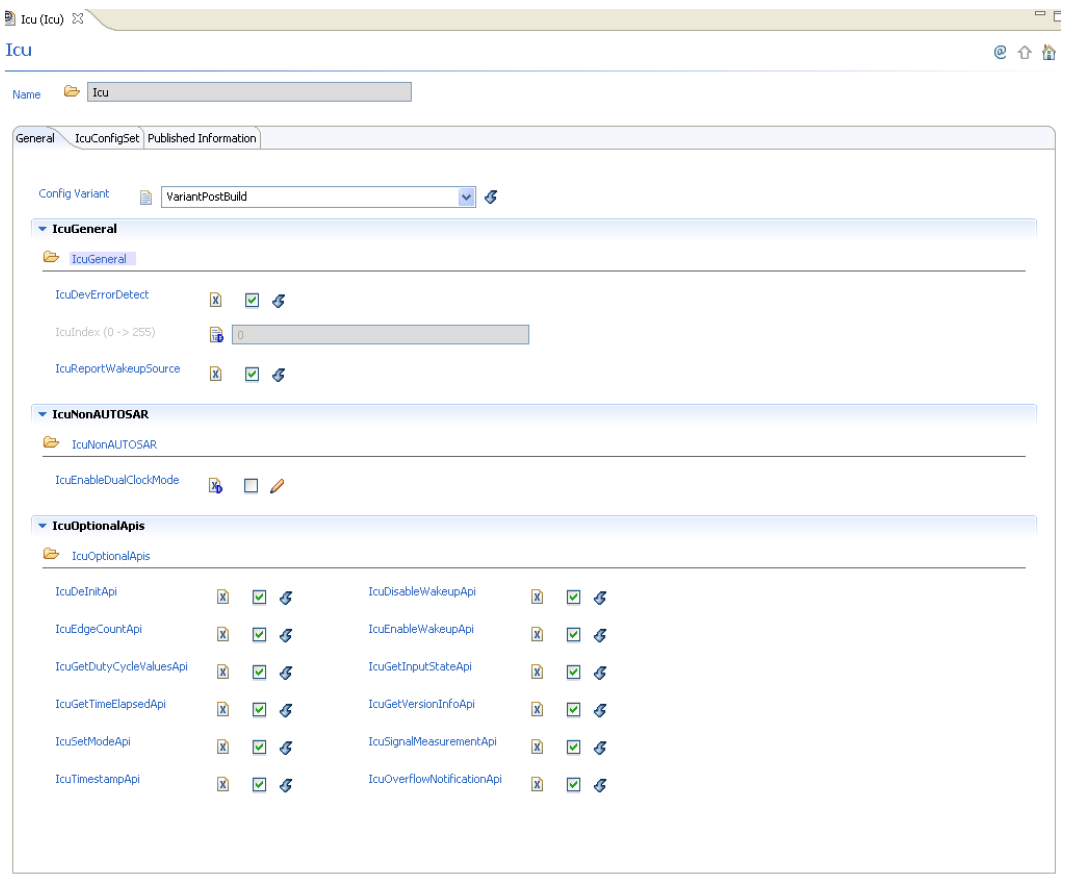

Figure 3-1. Tresos Plugin snapshot for IcuGeneral form

NOTE

If pre-compile variant is preferred,

- IMPLEMENTATION_CONFIG_VARIANT in Tresos GUI should be selected as “VariantPreCompile” as shown below

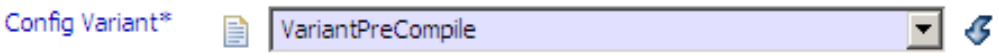

Figure 3-2. Tresos Plugin snapshot for IcuConfigurationVariant form

The following picture gives an example of configuration screen for all ICU configuration set.

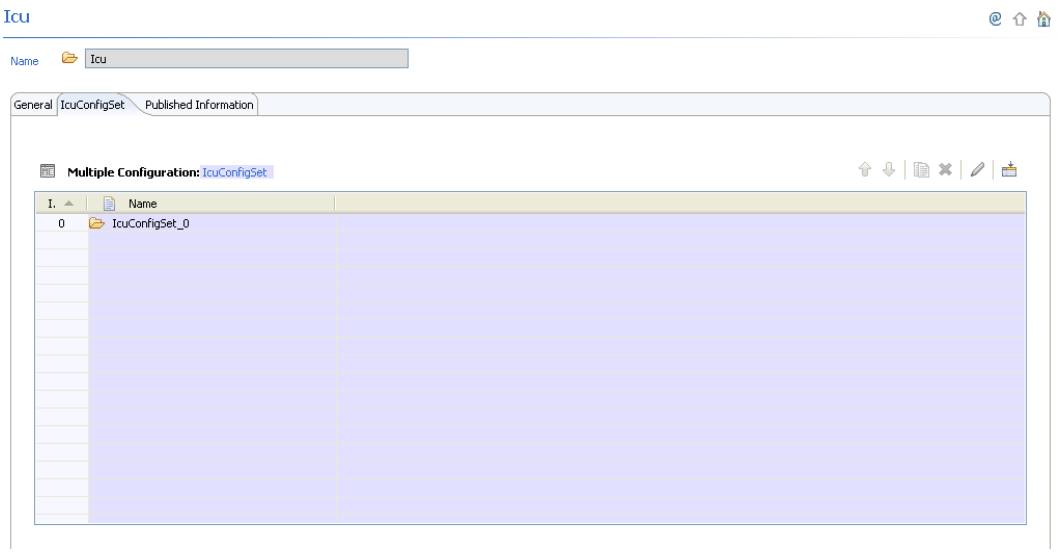

Figure 3-3. Tresos Plugin snapshot for IcuConfigsSet form

The following picture gives an example of configuration screen for General configuration for a specific set

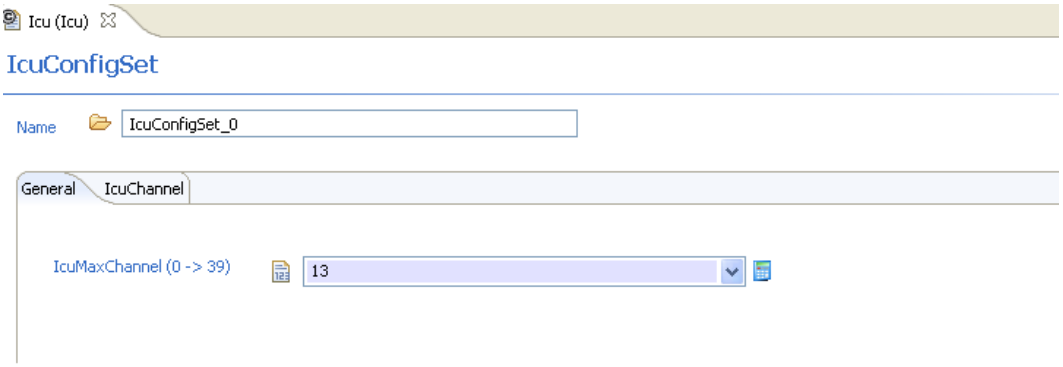

Figure 3-4. Tresos Plugin snapshot for Icu_MaxChannel form for a specific set

The following picture gives an example for IcuChannel container

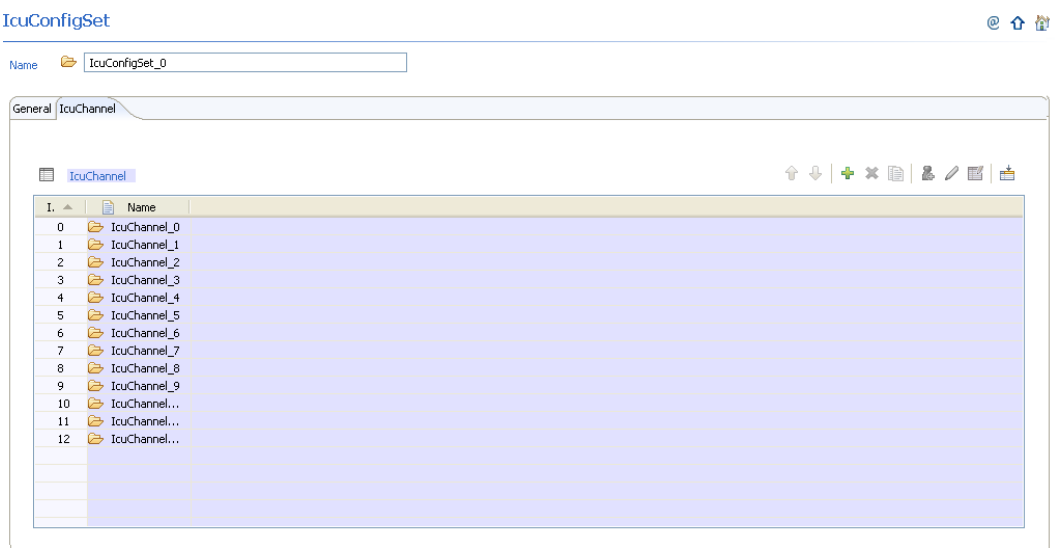

Figure 3-5. Tresos Plugin snapshot for Icu_Channel container

The following picture gives an example for a configuration screen of an Icu channel

IcuChannel

Name IcuChannel_0

General

IcuChannelId (0 -> 24) 0

IcuHwChannel EMIOS_0_CH_0

IcuEmiosFreeze

IcuEmiosPrescaler EMIOS_PRESCALER_DIVIDE_1

IcuEmiosPrescaler_Alternate EMIOS_PRESCALER_DIVIDE_1

IcuEmiosDigitalFilter EMIOS_DIGITAL_FILTER_BYPASSED

IcuEmiosBusSelect EMIOS_BUS_INTERNAL_COUNTER

IcuDefaultStartEdge ICU_RISING_EDGE

IcuMeasurementMode* ICU_MODE_SIGNAL_MEASUREMENT

IcuUserModeForDutycycle SAIC

IcuOverflowNotification NULL_PTR

IcuWakeupCapability

IcuEdgeCounterMeasurement

IcuEdgeCounterMeasurement

IcuSignalEdgeDetection

IcuSignalEdgeDetection

IcuSignalNotification NULL_PTR

IcuSignalMeasurement

IcuSignalMeasurement

IcuSignalMeasurementProperty ICU_DUTY_CYCLE

IcuTimestampMeasurement

IcuTimestampMeasurement

IcuTimestampMeasurementProperty ICU_LINEAR_BUFFER

IcuTimestampNotification NULL_PTR

IcuWakeup

IcuWakeup

IcuChannelWakeupInfo

Figure 3-6. Tresos Plugin snapshot for Icu_Channel form

3.11.2 Pre-Compile Parameters

Pre-Compile parameters, their possible values and meaning are described in the following text. Pre-Compile parameters are implemented as preprocessor defines.

NOTE

If pre-compile variant is preferred, then:

- IMPLEMENTATION_CONFIG_VARIANT in Tresos GUI should be selected as "VariantPreCompile" as shown below

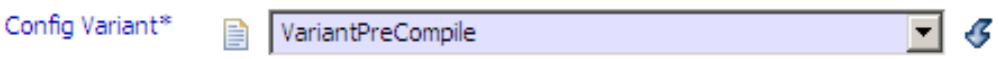

Figure 3-7. Tresos Plugin snapshot for IcuConfigurationVariant form

- Icu_Cfg.c should be compiled
- Icu_PBcfg.c should not be compiled

3.11.2.1 IcuMaxChannel

Table 3-36. IcuMaxChannel

Description	Number of configured icu channels
Class	Autosar Parameter
Range	0-40
Default	0
Source File	Icu_PBcfg.c

Source Representation	The highlighted portion of the following structure:
	`CONST(Icu_ConfigType, ICU_CONST) IcuConfigSet_0 =`
	`{`
	`  12, /* The number of channels configured*/`
	`  Icu_InitPBChannel_0,`
	`  Icu_InitPBChannelIndex_0,`
	`  Icu_ChannelId_0,`
	`  Icu_InitPBHWMap_0,`
	`  Icu_InitPBHWWKMap_0`
	`};`

3.11.2.2 IcuIndex

Table 3-37. IcuIndex

Description	This parameter Specifies the InstanceId of this module instance. If only one instance is present it shall have the Id 0. **NOTE:** In current implementation this parameter is not used.
Class	Autosar Parameter
Range	0-255
Default	0
Source File	Icu_Cfg.h
Source Representation	*#define ICU_INDEX <0-255>*
NOTE	*This parameter is not used in the current implementation*

3.11.2.3 IcuDevErrorDetect

Table 3-38. IcuDevErrorDetect

Description	Switches the Development Error Detection and Notification ON or OFF
Class	Autosar Parameter
Range	True,false
Default	True
Source File	Icu_Cfg.h
Source Representation	*#define ICU_DEV_ERROR_DETECT <STD_OFF, STD_ON>*

3.11.2.4 IcuReportWakeupSource

Table 3-39. IcuReportWakeupSource

Description	Switch for enabling Wakeup source reporting.
Class	Autosar Parameter
Range	True, False
Default	True
Source File	Icu_Cfg.h
Source Representation	*#define ICU_REPORT_WAKEUP_SOURCE <STD_OFF, STD_ON>*

3.11.2.5 IcuEnableDualClockMode

Table 3-40. IcuEnableDualClockMode

Description	Adds / removes the service Icu_SetClockMode () from the code.
Class	Non Autosar Parameter
Range	True,false
Default	False
Source File	Icu_Cfg.h
Source Representation	*#define ICU_DUAL_CLOCK_MODE <STD_OFF, STD_ON>*

3.11.2.6 IcuDeInitApi

Table 3-41. IcuDeInitApi

Description	Adds / removes the service Icu_DeInit() from the code
Class	Autosar Parameter
Range	True,false
Default	True
Source File	Icu_Cfg.h
Source Representation	*#define ICU_DE_INIT_API <STD_OFF, STD_ON>*

3.11.2.7 IcuDisableWakeupApi

Table 3-42. IcuDisableWakeupApi

Description	Adds / removes the service Icu_DisableWakeup () from the code.
Class	Autosar Parameter
Range	True, False
Default	True
Source File	Icu_Cfg.h
Source Representation	*#define ICU_DISABLE_WAKEUP_API<STD_OFF, STD_ON>*

3.11.2.8 IcuEdgeCountApi

Table 3-43. IcuEdgeCountApi

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

Table continues on the next page...

Table 3-43. IcuEdgeCountApi (continued)

Class	Autosar Parameter
Range	True , False
Default	True
Source File	Icu_Cfg.h
Source Representation	*#define ICU_EDGE_COUNT_API <STD_OFF, STD_ON>*

3.11.2.9 IcuEnableWakeupApi

Table 3-44. IcuEnableWakeupApi

Description	Adds / removes the service Icu_EnableWakeup () from the code.
Class	Autosar Parameter
Range	True,false
Default	True
Source File	Icu_Cfg.h
Source Representation	*#define ICU_ENABLE_WAKEUP_API <STD_OFF, STD_ON>*

3.11.2.10 IcuGetDutyCycleValuesApi

Table 3-45. IcuGetDutyCycleValuesApi

Description	Adds / removes the service Icu_GetDutyCycleValues () from the code. If IcuSignalMeasurementApi == OFF this switch should be set to OFF
Class	Autosar Parameter
Range	True, False
Default	True
Source File	Icu_Cfg.h
Source Representation	*#define ICU_GET_DUTY_CYCLE_VALUES_API <STD_OFF, STD_ON>*

3.11.2.11 IcuGetInputStateApi

Table 3-46. IcuGetInputStateApi

Description	Adds / removes the service Icu_GetInputState () from the code.
Class	Autosar Parameter
Range	True,false
Default	True

Table continues on the next page...

Table 3-46. IcuGetInputStateApi (continued)

Source File	Icu_Cfg.h
Source Representation	*#define ICU_GET_INPUT_STATE_API <STD_OFF, STD_ON>*

3.11.2.12 IcuGetTimeElapsedApi

Table 3-47. IcuGetTimeElapsedApi

Description	Adds / removes the service Icu_GetTimeElapsed () from the code. If IcuSignalMeasurementApi == OFF this switch should be set to OFF
Class	Autosar Parameter
Range	True,false
Default	True
Source File	Icu_Cfg.h
Source Representation	*#define ICU_GET_TIME_ELAPSED_API <STD_OFF, STD_ON>*

3.11.2.13 IcuGetVersionInfoApi

Table 3-48. IcuGetVersionInfoApi

Description	Adds / removes the service Icu_ GetVersionInfo () from the code.
Class	Autosar Parameter
Range	True,false
Default	True
Source File	Icu_Cfg.h
Source Representation	*#define ICU_GET_VERSION_INFO_API <STD_OFF, STD_ON>*

3.11.2.14 IcuSetModeApi

Table 3-49. IcuSetModeApi

Description	Adds / removes the service Icu_SetMode() from the code
Class	Autosar Parameter
Range	True,false
Default	True
Source File	Icu_Cfg.h
Source Representation	*#define ICU_SET_MODE_API <STD_OFF, STD_ON>*

3.11.2.15 IcuTimestampApi

Table 3-50. IcuTimestampApi

Description	Adds / removes all services related to the timestamping functionality as listed below from the code: Icu_StartTimestamp(), Icu_StopTimestamp(), Icu_GetTimestampIndex()
Class	Autosar Parameter
Range	True, False
Default	True
Source File	Icu_Cfg.h
Source Representation	*#define ICU_TIMESTAMP_API <STD_OFF, STD_ON>*

3.11.2.16 IcuSignalMeasurementApi

Table 3-51. IcuSignalMeasurementApi

Description	Adds / removes the services: Icu_StartSignalMeasurement () and Icu_StopSignalMeasurement ().
Class	Autosar Parameter
Range	True,false
Default	True
Source File	Icu_Cfg.h
Source Representation	*#define ICU_SIGNAL_MEASUREMENT_API <STD_OFF, STD_ON>*

3.11.2.17 IcuChannelId

Table 3-52. IcuChannelId

Description	Channel Id of the ICU channel.
Class	Autosar Parameter
Range	0-40
Default	0
Source File	Icu_PBcfg.c

Source Representation	The highlighted portion of the following structure:
	`CONST(Icu_ConfigType, ICU_CONST) IcuConfigSet_0 =`
	`{`
	`  1, /* The number of channels configured*/`
	`  Icu_InitPBChannel_0,`
	`  Icu_InitPBChannelIndex_0,`
	`  **Icu_ChannelId_0**,`
	`  Icu_InitPBHWMap_0,`
	`  Icu_InitPBHWWKMap_0`
	`};`

3.11.2.18 IcuHwChannel

Table 3-53. IcuHwChannel

Description	Channel Id of the ICU channel.
Class	Autosar Parameter
Range	EMIOS_0_CH[0-31], EMIOS_1_CH[0-31],IRQ[0-23], WKPU[0-28]
Default	EMIOS_0_CH_0
Source File	Icu_Cfg.c, Icu_PBCfg.c
Source Representation	***CONST(Icu_ChannelType,ICU_VAR) Icu_ChannelId_0[13]=*** ***{ ICU_IRQ_0,*** ***ICU_WKUP_4,*** ***ICU_IRQ_2,*** ***ICU_EMIOS_0_CH_9,*** ***ICU_EMIOS_0_CH_8,*** ***ICU_EMIOS_1_CH_16,*** ***ICU_EMIOS_1_CH_23,*** ***ICU_EMIOS_0_CH_16,*** ***ICU_EMIOS_0_CH_10,*** ***ICU_EMIOS_0_CH_11,*** ***ICU_EMIOS_1_CH_17,*** ***ICU_WKUP_10,*** ***ICU_WKUP_3 };***

3.11.2.19 icuoverflownotificationapi

Table 3-54. Icuoveflownotificationapi

Description	Adds / removes the service Overflow Notification functionality
Class	Autosar Parameter
Range	True,false
Default	False
Source File	Icu_Cfg.h
Source Representation	*#define ICU_OVERFLOW_NOTIFICATION_API <STD_OFF, STD_ON>*

3.11.3 Link Time parameters

N/A

3.11.4 Post - Build Parameters

Post-Build parameters, their possible values and their meaning are described in the following text. The Post-Build parameters are implemented as constant structures and arrays stored in flash memory of the ICU.

The configuration screen for ICU postbuild parameters in Tresos® Studio Configuration Tool is given below:

IcuChannel

Name IcuChannel_0

General

IcuChannelId (0 -> 24) 0

IcuHwChannel EMIOS_0_CH_0

IcuEmiosFreeze

IcuEmiosPrescaler EMIOS_PRESCALER_DIVIDE_1

IcuEmiosPrescaler_Alternate EMIOS_PRESCALER_DIVIDE_1

IcuEmiosDigitalFilter EMIOS_DIGITAL_FILTER_BYPASSED

IcuEmiosBusSelect EMIOS_BUS_INTERNAL_COUNTER

IcuDefaultStartEdge ICU_RISING_EDGE

IcuMeasurementMode* ICU_MODE_SIGNAL_MEASUREMENT

IcuUserModeForDutycycle SAIC

IcuOverflowNotification NULL_PTR

IcuWakeupCapability

IcuEdgeCounterMeasurement

IcuEdgeCounterMeasurement

IcuSignalEdgeDetection

IcuSignalEdgeDetection

IcuSignalNotification NULL_PTR

IcuSignalMeasurement

IcuSignalMeasurement

IcuSignalMeasurementProperty ICU_DUTY_CYCLE

IcuTimestampMeasurement

IcuTimestampMeasurement

IcuTimestampMeasurementProperty ICU_LINEAR_BUFFER

IcuTimestampNotification NULL_PTR

IcuWakeup

IcuWakeup

IcuChannelWakeupInfo

Figure 3-8. Tresos Plugin snapshot for Icu_Channel form

3.11.4.1 Icu_ParamType

Icu_ParamType is defined as a uint32. The Icu_ParamValue contains combined bit fields for initialization options, for different registers. Definitions are provided in Icu_cfg.h.

The table below depicts the Icu_ParamType values.

Table 3-55. Icu_ParamType Values

Definition	Description
ICU_FALLING_EDGE	Initialization option for eMIOS, IRQ and channels. It is the Default start edge when falling edge is selected.
ICU_RISING_EDGE	Initialization option for eMIOS, IRQ and channels. It is the Default start edge when rising edge is selected.
ICU_BOTH_EDGES	Initialization option for eMIOS, IRQ and channels. It is the Default start edge when both edges are selected.
ICU_WAKEUP_CAPABLE	Initialization option to specify the channel as wakeup capable. This works for eMIOS, IRQ and channels
EMIOS_BUS_A	Initialization option to specify that the channel runs off Timer Bus A. Hardware Dependent Definition. DO NOT CHANGE.
EMIOS_BUS_DIVERSE	Initialization option to specify that the channels 0 to 7 run off Timer Bus B, Channels 8 to 15 run off from Timer bus C. Hardware Dependent Definition. DO NOT CHANGE.
EMIOS_BUS_INTERNAL_COUNTER	Initialization option to specify that the channel runs off its own internal time base. Hardware Dependent Definition. DO NOT CHANGE.
EMIOS_DIGITAL_FILTER_BYPASSED	Initialization option to disable the filter. Hardware Dependent Definition. DO NOT CHANGE.
EMIOS_DIGITAL_FILTER_02	Initialization option to set the filter to 2 clocks. Hardware Dependent Definition. DO NOT CHANGE.
EMIOS_DIGITAL_FILTER_04	Initialization option to set the filter to 4 clocks. Hardware Dependent Definition. DO NOT CHANGE.
EMIOS_DIGITAL_FILTER_08	Initialization option to set the filter to 8 clocks. Hardware Dependent Definition. DO NOT CHANGE.
EMIOS_DIGITAL_FILTER_16	Initialization option to set the filter to 16 clocks. Hardware Dependent Definition. DO NOT CHANGE.
EMIOS_PRESCALER_DIVIDE_1	Initialization option to set the channel prescale value to 1. Only applies to channels configured to run off the internal counter bus. Hardware Dependent Definition. DO NOT CHANGE.
EMIOS_PRESCALER_DIVIDE_2	Initialization option to set the channel prescale value to 2. Only applies to channels configured to run off the internal counter bus. Hardware Dependent Definition. DO NOT CHANGE.
EMIOS_PRESCALER_DIVIDE_3	Initialization option to set the channel prescale value to 3. Only applies to channels configured to run off the internal counter bus. Hardware Dependent Definition. DO NOT CHANGE.
EMIOS_PRESCALER_DIVIDE_4	Initialization option to set the channel prescale value to 4. Only applies to channels configured to run off the internal counter bus. Hardware Dependent Definition. DO NOT CHANGE.

Table continues on the next page...

Table 3-55. Icu_ParamType Values (continued)

EMIOS_FREEZE_ENABLE	Initialization option to enable freeze mode for the channel. If this option is not specified, freeze mode will be disabled when Icu_Init is called for the channel. Hardware Dependent Definition. DO NOT CHANGE.
SIUL_INT_FILTER_ENABLE	Initialization option to enable IRQ filter
WKPU_INT_FILTER_ENABLE	Initialization option for analog filter on the corresponding interrupt pads to filter out glitches on the inputs
IcuEXT_ISR_IFMCDigitalFilter	Initialization option to configure the filter counter associated with each digital glitch filter.
WKPU_PULLUP_ENABLE	Initialization option to enable a pullup on the corresponding interrupt pads to pull an unconnected wakeup/interrupt input to a value of ‘1’

3.11.4.2 IcuEmiosFreeze

Table 3-56. IcuEmiosFreeze

Description	If selected eMIOS channel registers are freezed in debug mode.
Class	Implementation specific Non-Autosar parameter
Range	True,false
Default	True
Source File	Icu_PBcfg.c

Source Representation	The highlighted portion of the following structure:
	`CONST(Icu_ChannelConfigType, ICU_CONST) Icu_InitPBChannel_0[1] =`
	`{`
	`  {`
	`  ((ICU_RISING_EDGE (ls) ICU_EDGE_PARAM_SHIFT) |`
	`  (EMIOS_FREEZE_ENABLE (ls) ICU_EMIOS_FREEZE_PARAM_SHIFT)`
	`  (EMIOS_PRESCALER_DIVIDE_1 (ls)`
	`ICU_EMIOS_PRESCALER_PARAM_SHIFT) |(EMIOS_PRESCALER_DIVIDE_2 (ls)`
	`  ICU_EMIOS_PRESC_ALT_PARAM_SHIFT )`
	`  |(EMIOS_DIGITAL_FILTER_BYPASSED (ls)`
	`  ICU_EMIOS_DIGITAL_FILTER_PARAM_SHIFT) |`
	`  (EMIOS_BUS_INTERNAL_COUNTER (ls)`
	`  ICU_EMIOS_BUS_SELECT_PARAM_SHIFT)),`
	`  ICU_MODE_SIGNAL_MEASUREMENT,`
	`  ICU_DUTY_CYCLE,`
	`  NULL_PTR,`
	`  0U`
	`  }`
	`}`

3.11.4.3 IcuEmiosPrescaler

Table 3-57. IcuEmiosPrescaler

Description	If an eMIOS channel is being used, this configures the prescaler value for the specific channel.

Table continues on the next page...

Table 3-57. IcuEmiosPrescaler (continued)

Class	Implementation specific Non-Autosar parameter
Range	EMIOS_PRESCALER_DIVIDE_1, EMIOS_PRESCALER_DIVIDE_2, EMIOS_PRESCALER_DIVIDE_3, EMIOS_PRESCALER_DIVIDE_4
Default	EMIOS_PRESCALER_DIVIDE_1
Source File	Icu_PBcfg.c

Source Representation	The highlighted portion of the following structure:
	`CONST(Icu_ChannelConfigType, ICU_CONST) Icu_InitPBChannel_0[1] =`
	`{`
	`  {`
	`  ((ICU_RISING_EDGE (ls) ICU_EDGE_PARAM_SHIFT) |`
	`  (EMIOS_FREEZE_ENABLE (ls) ICU_EMIOS_FREEZE_PARAM_SHIFT)`
	`  (EMIOS_PRESCALER_DIVIDE_1 (ls)`
	`ICU_EMIOS_PRESCALER_PARAM_SHIFT) |(EMIOS_PRESCALER_DIVIDE_2 (ls)`
	`  ICU_EMIOS_PRESC_ALT_PARAM_SHIFT )`
	`  |(EMIOS_DIGITAL_FILTER_BYPASSED (ls)`
	`  ICU_EMIOS_DIGITAL_FILTER_PARAM_SHIFT) |`
	`  (EMIOS_BUS_INTERNAL_COUNTER (ls)`
	`  ICU_EMIOS_BUS_SELECT_PARAM_SHIFT)),`
	`  ICU_MODE_SIGNAL_MEASUREMENT,`
	`  ICU_DUTY_CYCLE,`
	`  NULL_PTR,`
	`  0U`
	`  }`
	`}`

3.11.4.4 IcuEmiosPrescaler_Alternate

Table 3-58. IcuEmiosPrescaler_Alternate

Description	this parameter configures the clock divider value for the internal prescaler of specific Unified Channel.
Class	Autosar Parameter
Range	EMIOS_PRESCALER_DIVIDE_1-EMIOS_PRESCALER_DIVIDE_4
Default	EMIOS_PRESCALER_DIVIDE_1
Source File	Icu_PBcfg.c

Table continues on the next page...

Table 3-58. IcuEmiosPrescaler_Alternate (continued)

Source Representation	The highlighted portion of the following structure:
	
	`CONST(Icu_ChannelConfigType, ICU_CONST) Icu_InitPBChannel_0[1] =`
	`{`
	`  {`
	`  (((Icu_ParamType)ICU_WAKEUP_CAPABLE (ls) ICU_WAKEUP_SHIFT) |`
	`  (ICU_RISING_EDGE (ls) ICU_EDGE_PARAM_SHIFT) |`
	`  (EMIOS_FREEZE_ENABLE (ls) ICU_EMIOS_FREEZE_PARAM_SHIFT)|`
	`  (EMIOS_PRESCALER_DIVIDE_1 (ls)`
	`ICU_EMIOS_PRESCALER_PARAM_SHIFT) |(EMIOS_PRESCALER_DIVIDE_2 (ls)`
	`  ICU_EMIOS_PRESC_ALT_PARAM_SHIFT )`
	`  |(EMIOS_DIGITAL_FILTER_BYPASSED (ls)`
	`  ICU_EMIOS_DIGITAL_FILTER_PARAM_SHIFT) |`
	`  (EMIOS_BUS_INTERNAL_COUNTER (ls)`
	`  ICU_EMIOS_BUS_SELECT_PARAM_SHIFT)),`
	`  ICU_MODE_SIGNAL_MEASUREMENT,`
	`  ICU_DUTY_CYCLE,`
	`  NULL_PTR,`
	`  0U`
	`  }`
	`}`

3.11.4.5 IcuEmiosDigitalFilter

Table 3-59. IcuEmiosDigitalFilter

Description	If a eMIOS channel is being used this option is active, possible values are: 0 (Bypassed), 2, 4, 8, 16 FLT_Clock periods.
Class	Implementation specific Non-Autosar parameter
Range	EMIOS_DIGITAL_FILTER_BYPASSED, EMIOS_DIGITAL_FILTER_02, EMIOS_DIGITAL_FILTER_04, EMIOS_DIGITAL_FILTER_08, EMIOS_DIGITAL_FILTER_16
Default	EMIOS_DIGITAL_FILTER_BYPASSED
Source File	Icu_PBcfg.c

Source Representation	The highlighted portion of the following structure:
	
	`CONST(Icu_ChannelConfigType, ICU_CONST) Icu_InitPBChannel_0[1] =`
	`{`
	
	`  {`
	`  ((ICU_RISING_EDGE (ls) ICU_EDGE_PARAM_SHIFT) |`
	`  (EMIOS_FREEZE_ENABLE (ls) ICU_EMIOS_FREEZE_PARAM_SHIFT)`
	`  (EMIOS_PRESCALER_DIVIDE_1 (ls)`
	`ICU_EMIOS_PRESCALER_PARAM_SHIFT) |(EMIOS_PRESCALER_DIVIDE_2 (ls)`
	`  ICU_EMIOS_PRESC_ALT_PARAM_SHIFT )`
	`  |(EMIOS_DIGITAL_FILTER_BYPASSED (ls)`
	`  ICU_EMIOS_DIGITAL_FILTER_PARAM_SHIFT) |`
	`  (EMIOS_BUS_INTERNAL_COUNTER (ls)`
	`  ICU_EMIOS_BUS_SELECT_PARAM_SHIFT)),`
	`  ICU_MODE_SIGNAL_MEASUREMENT,`
	`  ICU_DUTY_CYCLE,`
	`  NULL_PTR,`
	`  0U`
	`  }`
	`}`

3.11.4.6 IcuEmiosBusSelect

Table 3-60. IcuEmiosBusSelect

Description	Selects the counter used with the unified channel.
Class	Autosar Parameter
Range	EMIOS_BUS_A, EMIOS_BUS_DIVERSE,EMIOS_BUS_INTERNAL_COUNTER,
Default	EMIOS_BUS_INTERNAL_COUNTER (if applicable) or EMIOS_BUS_A
Source File	Icu_PBcfg.c

Source Representation	The highlighted portion of the following structure:
	
	`CONST(Icu_ChannelConfigType, ICU_CONST) Icu_InitPBChannel_0[1] =`
	`{`
	`  {`
	`  ((ICU_RISING_EDGE (ls) ICU_EDGE_PARAM_SHIFT) |`
	`  (EMIOS_FREEZE_ENABLE (ls) ICU_EMIOS_FREEZE_PARAM_SHIFT)`
	`  (EMIOS_PRESCALER_DIVIDE_1 (ls)`
	`ICU_EMIOS_PRESCALER_PARAM_SHIFT) |(EMIOS_PRESCALER_DIVIDE_2 (ls)`
	`  ICU_EMIOS_PRESC_ALT_PARAM_SHIFT )`
	`  |(EMIOS_DIGITAL_FILTER_BYPASSED (ls)`
	`  ICU_EMIOS_DIGITAL_FILTER_PARAM_SHIFT) |`
	`  (**EMIOS_BUS_INTERNAL_COUNTER (ls)**`
	`  **ICU_EMIOS_BUS_SELECT_PARAM_SHIFT**)),`
	`  ICU_MODE_SIGNAL_MEASUREMENT,`
	`  ICU_DUTY_CYCLE,`
	`  NULL_PTR,`
	`  0U`
	`  }`
	`}`

3.11.4.7 IcuDefaultStartEdge

Table 3-61. IcuDefaultStartEdge

Description	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 "SignalMeasurement" and the properties DutyCycle or Period are set, the edge configured here is used as Default Period Start Edge.
Class	Autosar Parameter
Range	IcuRisingEdge, IcuFallingEdge, IcuBothEdges
Default	IcuRisingEdge
Source File	Icu_PBcfg.c

Table continues on the next page...

Table 3-61. IcuDefaultStartEdge (continued)

Source Representation	The highlighted portion of the following structure:
	
	`CONST(Icu_ChannelConfigType, ICU_CONST) Icu_InitPBChannel_0[1] =`
	`{`
	
	`  {`
	`  ((ICU_RISING_EDGE (ls) ICU_EDGE_PARAM_SHIFT) |`
	`  (EMIOS_FREEZE_ENABLE (ls) ICU_EMIOS_FREEZE_PARAM_SHIFT)`
	`  (EMIOS_PRESCALER_DIVIDE_1 (ls)`
	`ICU_EMIOS_PRESCALER_PARAM_SHIFT)  |(EMIOS_PRESCALER_DIVIDE_2 (ls)`
	`  ICU_EMIOS_PRESC_ALT_PARAM_SHIFT )`
	`  |(EMIOS_DIGITAL_FILTER_BYPASSED (ls)`
	`  ICU_EMIOS_DIGITAL_FILTER_PARAM_SHIFT) |`
	`  (EMIOS_BUS_INTERNAL_COUNTER (ls)`
	`  ICU_EMIOS_BUS_SELECT_PARAM_SHIFT)),`
	`  ICU_MODE_SIGNAL_MEASUREMENT,`
	`  ICU_DUTY_CYCLE,`
	`  NULL_PTR,`
	`  0U`
	`  }`
	`}`

3.11.4.8 IcuMeasurementMode

Table 3-62. IcuMeasurementMode

Description	Configures the measurement mode of this channel.
Class	Autosar Parameter
Range	IcuSignalEdgeDetection, IcuEdgeCounter, IcuSignalMeasurement, IcuTimestamp
Default	IcuSignalEdgeDetection
Source File	Icu_PBcfg.c

Source Representation	The highlighted portion of the following structure:
	
	`CONST(Icu_ChannelConfigType, ICU_CONST) Icu_InitPBChannel_0[1] =`
	`{`
	
	`  {`
	`  ((ICU_RISING_EDGE (ls) ICU_EDGE_PARAM_SHIFT) |`
	`  (EMIOS_FREEZE_ENABLE (ls) ICU_EMIOS_FREEZE_PARAM_SHIFT)`
	`  (EMIOS_PRESCALER_DIVIDE_1 (ls)`
	`ICU_EMIOS_PRESCALER_PARAM_SHIFT)  |(EMIOS_PRESCALER_DIVIDE_2 (ls)`
	`  ICU_EMIOS_PRESC_ALT_PARAM_SHIFT )`
	`  |(EMIOS_DIGITAL_FILTER_BYPASSED (ls)`
	`  ICU_EMIOS_DIGITAL_FILTER_PARAM_SHIFT) |`
	`  (EMIOS_BUS_INTERNAL_COUNTER (ls)`
	`  ICU_EMIOS_BUS_SELECT_PARAM_SHIFT)),`
	`  **ICU_MODE_SIGNAL_MEASUREMENT,**`
	`  ICU_DUTY_CYCLE,`
	`  NULL_PTR,`
	`  0U`
	`  }`
	`}`

3.11.4.9 IcuUserModeForDutycycle

Table 3-63. IcuUserModeForDutycycle

Description	Selection of the signal measurement mode when IcuSignalMeasurementProperty is ICU_DUTY_CYCLE. **NOTE:** This parameter will be enabled in configuration only when IcuSignalMeasurementProperty is ICU_DUTY_CYCLE.
Class	Non Autosar Parameter
Range	IPWM, SAIC
Default	SAIC
Source File	Icu_PBcfg.c

Source Representation	The highlighted portion of the following structure:
	
	`CONST(Icu_ChannelConfigType, ICU_CONST) Icu_InitPBChannel_0[1] =`
	`{`
	`  {`
	
	`  (((Icu_ParamType)ICU_RISING_EDGE (ls) ICU_EDGE_PARAM_SHIFT) |`
	`  ((Icu_ParamType)EMIOS_UC_IPWM_MODE (ls)`
	`ICU_EMIOS_UC_MODE_PARAM_SHIFT)`
	`  | ((Icu_ParamType)EMIOS_FREEZE_ENABLE (ls)`
	`ICU_EMIOS_FREEZE_PARAM_SHIFT)|`
	`  ((Icu_ParamType)EMIOS_PRESCALER_DIVIDE_1 (ls)`
	`ICU_EMIOS_PRESCALER_PARAM_SHIFT)|`
	`  ((Icu_ParamType)EMIOS_DIGITAL_FILTER_BYPASSED (ls)`
	`ICU_EMIOS_DIG_FILTER_PARAM_SHIFT)|`
	`  ((Icu_ParamType)EMIOS_BUS_A (ls) ICU_EMIOS_BUS_SELECT_PARAM_SHIFT)),`
	`  ICU_MODE_SIGNAL_MEASUREMENT,`
	`  ICU_DUTY_CYCLE,`
	`  NULL_PTR,`
	`  0U`
	`  }`
	`}`

3.11.4.10 IcuWakeupCapability

Table 3-64. IcuWakeupCapability

Description	Boolean flag whether wake-up on this channel is supported or not.
Class	Autosar Parameter
Range	True,false
Default	True
Source File	Icu_PBcfg.c

Table continues on the next page...

Table 3-64. IcuWakeupCapability (continued)

Source Representation	The highlighted portion of the following structure:
	
	`CONST(Icu_ChannelConfigType, ICU_CONST) Icu_InitPBChannel_0[1] =`
	`{`
	
	`  {`
	`  (((Icu_ParamType)`**`ICU_WAKEUP_CAPABLE (ls) ICU_WAKEUP_SHIFT`**`) |`
	`  (ICU_RISING_EDGE (ls) ICU_EDGE_PARAM_SHIFT) |`
	`  (EMIOS_FREEZE_ENABLE (ls) ICU_EMIOS_FREEZE_PARAM_SHIFT)`
	`  (EMIOS_PRESCALER_DIVIDE_1 (ls)`
	`ICU_EMIOS_PRESCALER_PARAM_SHIFT) |(EMIOS_PRESCALER_DIVIDE_2 (ls)`
	`  ICU_EMIOS_PRESC_ALT_PARAM_SHIFT )`
	`  |(EMIOS_DIGITAL_FILTER_BYPASSED (ls)`
	`  ICU_EMIOS_DIGITAL_FILTER_PARAM_SHIFT) |`
	`  (EMIOS_BUS_INTERNAL_COUNTER (ls)`
	`  ICU_EMIOS_BUS_SELECT_PARAM_SHIFT)),`
	`  ICU_MODE_SIGNAL_MEASUREMENT,`
	`  ICU_DUTY_CYCLE,`
	`  NULL_PTR,`
	`  0U`
	`  }`
	`}`

3.11.4.11 IcuSignalNotification

Table 3-65. IcuSignalNotification

Description	Notification function for signal notification.
Class	Autosar Parameter
Range	NA
Default	"NULL"
Source File	Icu_PBcfg.c

Source Representation	The highlighted portion of the following structure:
	
	`CONST(Icu_ChannelConfigType, ICU_CONST) Icu_InitPBChannel_0[1] =`
	`{`
	
	`  {`
	`  (((Icu_ParamType)ICU_WAKEUP_CAPABLE (ls) ICU_WAKEUP_SHIFT) |`
	`  (ICU_RISING_EDGE (ls) ICU_EDGE_PARAM_SHIFT) |`
	`  (EMIOS_FREEZE_ENABLE (ls) ICU_EMIOS_FREEZE_PARAM_SHIFT)`
	`  (EMIOS_PRESCALER_DIVIDE_1 (ls)`
	`ICU_EMIOS_PRESCALER_PARAM_SHIFT) |(EMIOS_PRESCALER_DIVIDE_2 (ls)`
	`  ICU_EMIOS_PRESC_ALT_PARAM_SHIFT )`
	`  |(EMIOS_DIGITAL_FILTER_BYPASSED (ls)`
	`  ICU_EMIOS_DIGITAL_FILTER_PARAM_SHIFT) |`
	`  (EMIOS_BUS_INTERNAL_COUNTER (ls)`
	`  ICU_EMIOS_BUS_SELECT_PARAM_SHIFT)),`
	`  ICU_MODE_SIGNAL_EDGE_DETECT,`
	`  ICU_DUTY_CYCLE,`
	`  `**`EMIOS_0_CH_0_Notification`**`,`
	`  0U`
	`  }`
	`}`

3.11.4.12 IcuSignalMeasurementProperty

Table 3-66. IcuSignalMeasurementProperty

Description	Configures the property that could be measured in case the mode is "IcuSignalMeasurement".This property can not be changed during runtime.
Class	Autosar Parameter
Range	IcuPeriodTime, IcuDutyCycle, IcuHighTime, IcuLowTime
Default	IcuDutyCycle
Source File	Icu_PBcfg.c

Source Representation	The highlighted portion of the following structure:
	
	`CONST(Icu_ChannelConfigType, ICU_CONST) Icu_InitPBChannel_0[1] =`
	`{`
	
	`  {`
	`  ((ICU_RISING_EDGE (ls) ICU_EDGE_PARAM_SHIFT) |`
	`  (EMIOS_FREEZE_ENABLE (ls) ICU_EMIOS_FREEZE_PARAM_SHIFT)`
	`  (EMIOS_PRESCALER_DIVIDE_1 (ls)`
	`ICU_EMIOS_PRESCALER_PARAM_SHIFT) |(EMIOS_PRESCALER_DIVIDE_2 (ls)`
	`  ICU_EMIOS_PRESC_ALT_PARAM_SHIFT )`
	`  |(EMIOS_DIGITAL_FILTER_BYPASSED (ls)`
	`  ICU_EMIOS_DIGITAL_FILTER_PARAM_SHIFT) |`
	`  (EMIOS_BUS_INTERNAL_COUNTER (ls)`
	`  ICU_EMIOS_BUS_SELECT_PARAM_SHIFT)),`
	`  ICU_MODE_SIGNAL_MEASUREMENT,`
	`  **ICU_DUTY_CYCLE**,`
	`  NULL_PTR,`
	`  0U`
	`  }`
	`}`

3.11.4.13 IcuTimestampMeasurementProperty

Table 3-67. IcuTimestampMeasurementProperty

Description	Configures the handling of the buffer in case the mode is "Timestamp"
Class	Autosar Parameter
Range	IcuLinearBuffer, IcuCircularBuffer
Default	IcuLinearBuffer
Source File	Icu_PBcfg.c

Table continues on the next page...

Table 3-67. IcuTimestampMeasurementProperty (continued)

Source Representation	The highlighted portion of the following structure:
	
	`CONST(Icu_ChannelConfigType, ICU_CONST) Icu_InitPBChannel_0[1] =`
	`{`
	
	`  {`
	`  (((Icu_ParamType)ICU_WAKEUP_CAPABLE (ls) ICU_WAKEUP_SHIFT) |`
	`  (ICU_RISING_EDGE (ls) ICU_EDGE_PARAM_SHIFT) |`
	`  (EMIOS_FREEZE_ENABLE (ls) ICU_EMIOS_FREEZE_PARAM_SHIFT)`
	`  (EMIOS_PRESCALER_DIVIDE_1 (ls)`
	`ICU_EMIOS_PRESCALER_PARAM_SHIFT) |(EMIOS_PRESCALER_DIVIDE_2 (ls)`
	`  ICU_EMIOS_PRESC_ALT_PARAM_SHIFT )`
	`  |(EMIOS_DIGITAL_FILTER_BYPASSED (ls)`
	`  ICU_EMIOS_DIGITAL_FILTER_PARAM_SHIFT) |`
	`  (EMIOS_BUS_INTERNAL_COUNTER (ls)`
	`  ICU_EMIOS_BUS_SELECT_PARAM_SHIFT)),`
	`  ICU_MODE_TIME_STAMP,`
	`  **ICU_CIRCULAR_BUFFER**,`
	`  EMIOS_0_CH_0_Notification,`
	`  0U`
	`  }`
	`}`

3.11.4.14 IcuTimestampNotification

Table 3-68. IcuTimestampNotification

Description	Notification function if the number of requested timestamps (Notification interval > 0) are acquired.
Class	Autosar Parameter
Range	NA
Default	NULL_PTR
Source File	Icu_PBcfg.c

Source Representation	The highlighted portion of the following structure:
	
	`CONST(Icu_ChannelConfigType, ICU_CONST) Icu_InitPBChannel_0[1] =`
	`{`
	
	`  {`
	`  (((Icu_ParamType)ICU_WAKEUP_CAPABLE (ls) ICU_WAKEUP_SHIFT) |`
	`  (ICU_RISING_EDGE (ls) ICU_EDGE_PARAM_SHIFT) |`
	`  (EMIOS_FREEZE_ENABLE (ls) ICU_EMIOS_FREEZE_PARAM_SHIFT)`
	`  (EMIOS_PRESCALER_DIVIDE_1 (ls)`
	`ICU_EMIOS_PRESCALER_PARAM_SHIFT) |(EMIOS_PRESCALER_DIVIDE_2 (ls)`
	`  ICU_EMIOS_PRESC_ALT_PARAM_SHIFT )`
	`  |(EMIOS_DIGITAL_FILTER_BYPASSED (ls)`
	`  ICU_EMIOS_DIGITAL_FILTER_PARAM_SHIFT) |`
	`  (EMIOS_BUS_INTERNAL_COUNTER (ls)`
	`  ICU_EMIOS_BUS_SELECT_PARAM_SHIFT)),`
	`  ICU_MODE_TIME_STAMP,`
	`  ICU_CIRCULAR_BUFFER,`
	`  **Timestamp1_Notification**,`
	`  0U`
	`  }`
	`}`

3.11.4.15 IcuChannelWakeupInfo

Table 3-69. IcuChannelWakeupInfo

Description	If the wakeup capability is true the wakeup source referenced is transmitted to the ECU State Manager (EcuM).
Class	Autosar Parameter
Range	NA
Default	NA
Source File	Icu_PBcfg.c

Source Representation	The highlighted portion of the following structure:
	`CONST(Icu_ChannelConfigType, ICU_CONST) Icu_InitPBChannel_0[1] =`
	`{`
	`  {`
	`  (((Icu_ParamType)ICU_WAKEUP_CAPABLE (ls) ICU_WAKEUP_SHIFT) |`
	`  (ICU_RISING_EDGE (ls) ICU_EDGE_PARAM_SHIFT) |`
	`  (EMIOS_FREEZE_ENABLE (ls) ICU_EMIOS_FREEZE_PARAM_SHIFT)`
	`  (EMIOS_PRESCALER_DIVIDE_1 (ls)`
	`ICU_EMIOS_PRESCALER_PARAM_SHIFT) |(EMIOS_PRESCALER_DIVIDE_2 (ls)`
	`  ICU_EMIOS_PRESC_ALT_PARAM_SHIFT )`
	`  |(EMIOS_DIGITAL_FILTER_BYPASSED (ls)`
	`  ICU_EMIOS_DIGITAL_FILTER_PARAM_SHIFT) |`
	`  (EMIOS_BUS_INTERNAL_COUNTER (ls)`
	`  ICU_EMIOS_BUS_SELECT_PARAM_SHIFT)),`
	`  ICU_MODE_TIME_STAMP,`
	`  ICU_CIRCULAR_BUFFER,`
	`  Icu_SignalNotification_IRQ_1_User,`
	`  **0U**`
	`  }`
	`}`

3.11.4.16 IcuOverflowNotification

Table 3-70. IcuOverflowNotification

Description	Icu Overflow Notification Handle. 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

Class	Autosar Parameter
Range	NA
Default	"NULL_PTR"
Source File	Icu_PBcfg.c

Table continues on the next page...

Table 3-70. IcuOverflowNotification (continued)

Source Representation	The highlighted portion of the following structure:
	
	`CONST(Icu_ChannelConfigType, ICU_CONST) Icu_InitPBChannel_0[1] =`
	`{`
	
	`  {`
	`  ((ICU_RISING_EDGE (ls) ICU_EDGE_PARAM_SHIFT) |`
	`  (EMIOS_FREEZE_ENABLE (ls) ICU_EMIOS_FREEZE_PARAM_SHIFT)`
	`  (EMIOS_PRESCALER_DIVIDE_1 (ls)`
	`ICU_EMIOS_PRESCALER_PARAM_SHIFT)  |(EMIOS_PRESCALER_DIVIDE_2 (ls)`
	`  ICU_EMIOS_PRESC_ALT_PARAM_SHIFT )`
	`  |(EMIOS_DIGITAL_FILTER_BYPASSED (ls)`
	`  ICU_EMIOS_DIGITAL_FILTER_PARAM_SHIFT) |`
	`  (EMIOS_BUS_INTERNAL_COUNTER (ls)`
	`  ICU_EMIOS_BUS_SELECT_PARAM_SHIFT)),`
	`  ICU_MODE_SIGNAL_MEASUREMENT,`
	`  ICU_DUTY_CYCLE,`
	`  NULL_PTR,`
	`  **0U**`
	`  }`
	`}`

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
1-800-441-2447 or +1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Document Number: UM14ICUASR3.0R2.0.0
Rev. 2.0