
User Manual

for MPC574XG ICU Driver

Document Number: UM35ICUASR4.2 Rev0002RTM1.0.0
Rev. 5.0.0





Contents

Section number	Title	Page
Chapter 1		
Revision History		
Chapter 2		
Introduction		
2.1	Supported Derivatives.....	15
2.2	Overview.....	16
2.3	About this Manual.....	16
2.4	Acronyms and Definitions.....	17
2.5	Reference List.....	17
Chapter 3		
Driver		
3.1	Requirements.....	19
3.2	Driver Design Summary.....	19
3.3	Hardware Resources.....	20
3.4	Driver usage and configuration tips.....	20
3.5	Driver limitations.....	20
3.6	Hardware Channel Availability.....	20
3.7	Additional Requirements and Deviations.....	21
3.8	eMIOS specific implementation details.....	24
3.9	Runtime Errors.....	27
3.10	Software specification.....	27
3.10.1	Define Reference.....	27
3.10.1.1	Define ICU_VENDOR_ID.....	27
3.10.1.2	Define ICU_AR_RELEASE_MAJOR_VERSION.....	27
3.10.1.3	Define ICU_AR_RELEASE_MINOR_VERSION.....	28
3.10.1.4	Define ICU_AR_RELEASE_REVISION_VERSION.....	28
3.10.1.5	Define ICU_SW_MAJOR_VERSION.....	28
3.10.1.6	Define ICU_SW_MINOR_VERSION.....	29

Section number	Title	Page
3.10.1.7	Define ICU_SW_PATCH_VERSION.....	29
3.10.1.8	Define ICU_CHECKWAKEUP_ID.....	29
3.10.1.9	Define ICU_DEINIT_ID.....	30
3.10.1.10	Define ICU_DISABLEEDGECOUNT_ID.....	30
3.10.1.11	Define ICU_DISABLEEDGEDETECTION_ID.....	30
3.10.1.12	Define ICU_DISABLENOTIFICATION_ID.....	31
3.10.1.13	Define ICU_DISABLEWAKEUP_ID.....	31
3.10.1.14	Define ICU_ENABLEEDGECOUNT_ID.....	31
3.10.1.15	Define ICU_ENABLEEDGEDETECTION_ID.....	32
3.10.1.16	Define ICU_ENABLENOTIFICATION_ID.....	32
3.10.1.17	Define ICU_ENABLEWAKEUP_ID.....	32
3.10.1.18	Define ICU_GET_INPUT_LEVEL_ID.....	33
3.10.1.19	Define ICU_GETDUTYCYCLEVALUES_ID.....	33
3.10.1.20	Define ICU_GETEDGENUMBERS_ID.....	34
3.10.1.21	Define ICU_GETINPUTSTATE_ID.....	34
3.10.1.22	Define ICU_GETTIMEELAPSED_ID.....	34
3.10.1.23	Define ICU_GETTIMESTAMPINDEX_ID.....	35
3.10.1.24	Define ICU_GETVERSIONINFO_ID.....	35
3.10.1.25	Define ICU_INIT_ID.....	35
3.10.1.26	Define ICU_MODULE_ID.....	36
3.10.1.27	Define ICU_RESETEdgeCOUNT_ID.....	36
3.10.1.28	Define ICU_SETACTIVATIONCONDITION_ID.....	36
3.10.1.29	Define ICU_SETMODE_ID.....	37
3.10.1.30	Define ICU_STARTSIGNALMEASUREMENT_ID.....	37
3.10.1.31	Define ICU_STARTTIMESTAMP_ID.....	37
3.10.1.32	Define ICU_STOPSIGNALMEASUREMENT_ID.....	38
3.10.1.33	Define ICU_STOPTIMESTAMP_ID.....	38
3.10.1.34	Define ICU_SET_CLOCK_MODE_ID.....	38
3.10.1.35	Define ICU_GET_PULSE_WIDTH_ID.....	39

Section number	Title	Page
3.10.1.36	Define ICU_GET_CAPTURE_REGISTER_VALUE_ID.....	39
3.10.1.37	Define ICU_E_ALREADY_INITIALIZED.....	39
3.10.1.38	Define ICU_E_BUSY_OPERATION.....	40
3.10.1.39	Define ICU_E_EDGECOUNTING_OVERFLOW.....	40
3.10.1.40	Define ICU_E_MEASUREMENT_OVERFLOW.....	40
3.10.1.41	Define ICU_E_NOT_STARTED.....	41
3.10.1.42	Define ICU_E_PARAM_ACTIVATION.....	41
3.10.1.43	Define ICU_E_PARAM_POINTER.....	41
3.10.1.44	Define ICU_E_PARAM_BUFFER_SIZE.....	42
3.10.1.45	Define ICU_E_PARAM_CHANNEL.....	42
3.10.1.46	Define ICU_E_PARAM_MODE.....	42
3.10.1.47	Define ICU_E_PARAM_NOTIFY_INTERVAL.....	43
3.10.1.48	Define ICU_E_PARAM_VINFO.....	43
3.10.1.49	Define ICU_E_TIMESTAMP_OVERFLOW.....	43
3.10.1.50	Define ICU_E_UNINIT.....	44
3.10.1.51	Define ICU_E_INVALID_MODE.....	44
3.10.1.52	Define ICU_E_FORBIDEN_MODE.....	44
3.10.1.53	Define ICU_E_PARAM_CLOCK_MODE.....	45
3.10.1.54	Define ICU_E_INIT_FAILED.....	45
3.10.1.55	Define ICU_GET_VERSION_INFO_API.....	45
3.10.1.56	Define ICU_DE_INIT_API.....	46
3.10.1.57	Define ICU_SET_MODE_API.....	46
3.10.1.58	Define ICU_DISABLE_WAKEUP_API.....	46
3.10.1.59	Define ICU_ENABLE_WAKEUP_API.....	47
3.10.1.60	Define ICU_TIMESTAMP_API.....	47
3.10.1.61	Define ICU_EDGE_COUNT_API.....	47
3.10.1.62	Define ICU_GET_TIME_ELAPSED_API.....	48
3.10.1.63	Define ICU_GET_DUTY_CYCLE_VALUES_API.....	48
3.10.1.64	Define ICU_GET_INPUT_STATE_API.....	49

Section number	Title	Page
3.10.1.65	Define ICU_SIGNAL_MEASUREMENT_API.....	49
3.10.1.66	Define ICU_WAKEUP_FUNCTIONALITY_API.....	49
3.10.1.67	Define ICU_EDGE_DETECT_API.....	50
3.10.1.68	Define ICU_OVERFLOW_NOTIFICATION_API.....	50
3.10.1.69	Define ICU_GET_INPUT_LEVEL_API.....	50
3.10.1.70	Define ICU_DUAL_CLOCK_MODE_API.....	51
3.10.1.71	Define ICU_TIMESTAMP_USES_DMA.....	51
3.10.1.72	Define ICU_SIGNALMEASUREMENT_USES_DMA.....	51
3.10.1.73	Define ICU_GET_PULSE_WIDTH_API.....	52
3.10.1.74	Define ICU_CAPTUREREGISTER_API.....	52
3.10.1.75	Define ICU_REPORT_WAKEUP_SOURCE.....	52
3.10.1.76	Define ICU_DEV_ERROR_DETECT.....	53
3.10.2	Enum Reference.....	53
3.10.2.1	Enumeration Icu_ActivationType.....	53
3.10.2.2	Enumeration Icu_InputStateType.....	54
3.10.2.3	Enumeration Icu_MeasurementModeType.....	54
3.10.2.4	Enumeration Icu_ModeType.....	54
3.10.2.5	Enumeration Icu_SignalMeasurementPropertyType.....	55
3.10.2.6	Enumeration Icu_TimestampBufferType.....	55
3.10.2.7	Enumeration Icu_SelectPrescalerType.....	56
3.10.2.8	Enumeration Icu_LevelType.....	56
3.10.2.9	Enumeration Icu_Siul2_ActivationType.....	56
3.10.2.10	Enumeration Icu_Siul2_SelectPrescalerType.....	57
3.10.3	Function Reference.....	57
3.10.3.1	Function Icu_Init.....	57
3.10.3.2	Function Icu_DeInit.....	58
3.10.3.3	Function Icu_SetMode.....	58
3.10.3.4	Function Icu_GetVersionInfo.....	59
3.10.3.5	Function Icu_CheckWakeup.....	60

Section number	Title	Page
3.10.3.6	Function Icu_DisableWakeup.....	60
3.10.3.7	Function Icu_EnableWakeup.....	61
3.10.3.8	Function Icu_DisableEdgeCount.....	62
3.10.3.9	Function Icu_EnableEdgeCount.....	62
3.10.3.10	Function Icu_ResetEdgeCount.....	63
3.10.3.11	Function Icu_GetEdgeNumbers.....	63
3.10.3.12	Function Icu_DisableEdgeDetection.....	64
3.10.3.13	Function Icu_EnableEdgeDetection.....	64
3.10.3.14	Function Icu_DisableNotification.....	65
3.10.3.15	Function Icu_EnableNotification.....	66
3.10.3.16	Function Icu_SetActivationCondition.....	66
3.10.3.17	Function Icu_GetInputLevel.....	67
3.10.3.18	Function Icu_GetInputState.....	67
3.10.3.19	Function Icu_StartSignalMeasurement.....	68
3.10.3.20	Function Icu_StopSignalMeasurement.....	69
3.10.3.21	Function Icu_GetDutyCycleValues.....	70
3.10.3.22	Function Icu_GetTimeElapsed.....	70
3.10.3.23	Function Icu_GetTimestampIndex.....	71
3.10.3.24	Function Icu_StartTimestamp.....	72
3.10.3.25	Function Icu_StopTimestamp.....	72
3.10.3.26	Function Icu_SetClockMode.....	73
3.10.3.27	Function Icu_GetCaptureRegisterValue.....	73
3.10.3.28	Function Icu_GetPulseWidth.....	74
3.10.4	Structs Reference.....	75
3.10.4.1	Structure Icu_ChannelConfigType.....	75
3.10.4.2	Structure Icu_ConfigType.....	76
3.10.4.3	Structure Icu_IpConfigType.....	77
3.10.4.4	Structure Icu_IpChannelConfigType.....	78
3.10.4.5	Structure Icu_eMios_IpConfigType.....	78

Section number	Title	Page
3.10.4.6	Structure Icu_eMios_ChannelConfigType.....	78
3.10.4.7	Structure Icu_Siul2_IpConfigType.....	79
3.10.4.8	Structure Icu_Siul2_ChannelConfigType.....	80
3.10.4.9	Structure Icu_Wkpu_IpConfigType.....	80
3.10.4.10	Structure Icu_Wkpu_ChannelConfigType.....	80
3.10.4.11	Structure Icu_DutyCycleType.....	81
3.10.5	Types Reference.....	82
3.10.5.1	Typedef Icu_ChannelType.....	82
3.10.5.2	Typedef Icu_EdgeNumberType.....	83
3.10.5.3	Typedef Icu_IndexType.....	83
3.10.5.4	Typedef Icu_NotifyType.....	83
3.10.5.5	Typedef Icu_ParamType.....	83
3.10.5.6	Typedef Icu_ValueType.....	84
3.10.5.7	Typedef Icu_WakeupValueType.....	84
3.10.5.8	Typedef Icu_MeasurementSubModeType.....	84
3.10.5.9	Typedef Icu_GlobalConfigType.....	84
3.10.6	Variables	85
3.11	Symbolic Names Disclaimer.....	87

Chapter 4 Tresos Configuration Plug-in

4.1	Configuration elements of Icu.....	89
4.2	Form IMPLEMENTATION_CONFIG_VARIANT.....	89
4.3	Form IcuGeneral.....	90
4.3.1	IcuDevErrorDetect (IcuGeneral).....	90
4.3.2	IcuIndex (IcuGeneral).....	90
4.3.3	IcuReportWakeupSource (IcuGeneral).....	91
4.3.4	IcuEnableUserModeSupport (IcuGeneral).....	91
4.4	Form IcuNonAUTOSAR.....	92
4.4.1	IcuOverflowNotificationApi (IcuNonAUTOSAR).....	92

Section number	Title	Page
4.4.2	IcuGetInputLevelApi (IcuNonAUTOSAR).....	93
4.4.3	IcuRegisterLockingMode (IcuNonAUTOSAR).....	93
4.4.4	IcuEnableDualClockMode (IcuNonAUTOSAR).....	94
4.4.5	IcuSupportSAICModeApi (IcuNonAUTOSAR).....	94
4.5	Form IcuOptionalApis.....	94
4.5.1	IcuDeInitApi (IcuOptionalApis).....	95
4.5.2	IcuDisableWakeupApi (IcuOptionalApis).....	95
4.5.3	IcuEdgeCountApi (IcuOptionalApis).....	96
4.5.4	IcuEnableWakeupApi (IcuOptionalApis).....	96
4.5.5	IcuGetDutyCycleValuesApi (IcuOptionalApis).....	97
4.5.6	IcuGetInputStateApi (IcuOptionalApis).....	97
4.5.7	IcuGetTimeElapsedApi (IcuOptionalApis).....	97
4.5.8	IcuGetVersionInfoApi (IcuOptionalApis).....	98
4.5.9	IcuSetModeApi (IcuOptionalApis).....	98
4.5.10	IcuSignalMeasurementApi (IcuOptionalApis).....	99
4.5.11	IcuTimestampApi (IcuOptionalApis).....	99
4.5.12	IcuWakeupFunctionalityApi (IcuOptionalApis).....	100
4.5.13	IcuEdgeDetectApi (IcuOptionalApis).....	100
4.6	Form CommonPublishedInformation.....	100
4.6.1	ArReleaseMajorVersion (CommonPublishedInformation).....	101
4.6.2	ArReleaseMinorVersion (CommonPublishedInformation).....	101
4.6.3	ArReleaseRevisionVersion (CommonPublishedInformation).....	102
4.6.4	ModuleId (CommonPublishedInformation).....	102
4.6.5	SwMajorVersion (CommonPublishedInformation).....	103
4.6.6	SwMinorVersion (CommonPublishedInformation).....	103
4.6.7	SwPatchVersion (CommonPublishedInformation).....	104
4.6.8	VendorApiInfix (CommonPublishedInformation).....	104
4.6.9	VendorId (CommonPublishedInformation).....	105
4.7	Form IcuConfigSet.....	105

Section number	Title	Page
4.7.1	IcuMaxChannel (IcuConfigSet).....	105
4.7.2	Form IcuChannel.....	106
4.7.2.1	IcuChannelId (IcuChannel).....	106
4.7.2.2	IcuHwIP (IcuChannel).....	107
4.7.2.3	IcueMiosChannelRef (IcuChannel).....	107
4.7.2.4	IcuSiul2ChannelRef (IcuChannel).....	108
4.7.2.5	IcuWkupChannelRef (IcuChannel).....	108
4.7.2.6	IcuDMACChannelEnable (IcuChannel).....	108
4.7.2.7	IcuDefaultStartEdge (IcuChannel).....	108
4.7.2.8	IcuMeasurementMode (IcuChannel).....	109
4.7.2.9	IcuOverflowNotification (IcuChannel).....	110
4.7.2.10	IcuLockableChannel (IcuChannel).....	110
4.7.2.11	IcuWakeupCapability (IcuChannel).....	111
4.7.2.12	IcuDMACChannelRef (IcuChannel).....	111
4.7.2.13	Form IcuSignalEdgeDetection.....	111
4.7.2.13.1	IcuSignalNotification (IcuSignalEdgeDetection).....	112
4.7.2.14	Form IcuSignalMeasurement.....	112
4.7.2.14.1	IcuSignalMeasurementProperty (IcuSignalMeasurement).....	112
4.7.2.15	Form IcuTimestampMeasurement.....	113
4.7.2.15.1	IcuTimestampMeasurementProperty (IcuTimestampMeasurement).....	113
4.7.2.15.2	IcuTimestampNotification (IcuTimestampMeasurement).....	114
4.7.2.16	Form IcuWakeup.....	114
4.7.2.16.1	IcuChannelWakeupInfo (IcuWakeup).....	114
4.7.3	Form IcueMios.....	115
4.7.3.1	IcueMiosModule (IcueMios).....	115
4.7.3.2	Form IcueMiosChannel.....	115
4.7.3.2.1	IcueMiosChannel (IcueMiosChannel).....	116
4.7.3.2.2	IcuEmiosFreeze (IcueMiosChannel).....	116
4.7.3.2.3	IcuEmiosPrescaler (IcueMiosChannel).....	116

Section number	Title	Page
4.7.3.2.4	IcuEmiosPrescaler_Alternate(IcucMiosChannel).....	117
4.7.3.2.5	IcuEmiosDigitalFilter(IcucMiosChannel).....	117
4.7.3.2.6	IcuEmiosBusSelect(IcucMiosChannel).....	118
4.7.3.2.7	IcuUserModeForDutycycle (IcucMiosChannel).....	119
4.7.3.2.8	IcuSignalMeasureWithoutInterrupt (IcucMiosChannel).....	119
4.7.3.3	Form IcucMiosMasterBus.....	120
4.7.3.3.1	eMiosMasterBus (IcucMiosMasterBus).....	120
4.7.3.3.2	MasterBusPrescaler (IcucMiosMasterBus).....	120
4.7.3.3.3	MasterBusPrescaler_Alternate (IcucMiosMasterBus).....	121
4.7.4	Form IcuSiul2.....	121
4.7.4.1	ICU External Interrupt Filter Clock Prescaler (IcuSiul2).....	122
4.7.4.2	ICU External Alternate Interrupt Filter Clock Prescaler (IcuSiul2).....	122
4.7.4.3	Form IcuSiul2Channel.....	123
4.7.4.3.1	IcuSiul2Channel (IcuSiul2Channel).....	123
4.7.4.3.2	Icu_EXT_ISR_IFERDigitalFilter (IcuSiul2Channel).....	123
4.7.4.3.3	Icu_EXT_ISR_IFMCDigitalFilter (IcuSiul2Channel).....	124
4.7.5	Form IcuWkpu.....	124
4.7.5.1	IcuWkpuChannel (IcuWkpu).....	125
4.7.5.2	IcuWKPU_ISR_WIPUER (IcuWkpu).....	125
4.7.5.3	Icu_EXT_ISR_WIFERDigitalFilter (IcuWkpu).....	125
4.7.6	Form IcuHwInterruptConfigList.....	126
4.7.6.1	ICU Peripheral ISR Name (IcuHwInterruptConfigList).....	126
4.7.6.2	IcuIsrEnable(IcuHwInterruptConfigList).....	127



Chapter 1

Revision History

Table 1-1. Revision History

Revision	Date	Author	Description
1.0.0	18/08/2014	Son Nguyen	User Manual for MPC574XG - 0.9.0 Release
2.0.0	24/04/2015	Lam Tran	User Manual for MPC574XG - RTM 1.0.0 Release
3.0.0	10/07/2015	Lam Tran	User Manual for CALPYSO - RTM 1.0.1 Release
4.0.0	12/08/2016	My Le	User Manual for CALPYSO - RTM 1.0.2 Release
5.0.0	17/02/2017	My Le	User Manual for CALPYSO AUTOSAR 4.2.2-RTM 1.0.0 Release



Chapter 2

Introduction

This User Manual describes NXP Semiconductor AUTOSAR microcontroller abstraction layer (MCAL) Input Capture Unit (ICU) for MPC574XG microcontroller.

AUTOSAR ICU driver configuration parameters and deviations from the specification are described in the 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 Semiconductor .

Table 2-1. MPC574XG Derivatives

NXP Semiconductor	MPC5748G_LQFP176, MPC5748G_MAPBGA256, MPC5748G_MAPBGA324, MPC5747G_LQFP176, MPC5747G_MAPBGA256, MPC5747G_MAPBGA324, MPC5746G_LQFP176, MPC5746G_MAPBGA256, MPC5746G_MAPBGA324, MPC5748C_LQFP176, MPC5748C_MAPBGA256, MPC5748C_MAPBGA324, MPC5747C_LQFP176, MPC5747C_MAPBGA256, MPC5747C_MAPBGA324, MPC5746C_LQFP176, MPC5746C_MAPBGA256, MPC5746C_MAPBGA324, MPC5746C_MAPBGA100, MPC5745C_LQFP176, MPC5745C_MAPBGA256, MPC5745C_MAPBGA100, MPC5744C_LQFP176, MPC5744C_MAPBGA256,
-------------------	--

Table 2-1. MPC574XG Derivatives

	MPC5744C_MAPBGA100, MPC5746B_LQFP176, MPC5746B_MAPBGA256, MPC5746B_MAPBGA100, MPC5744B_LQFP176, MPC5744B_MAPBGA256, MPC5744B_MAPBGA100, MPC5745B_LQFP176, MPC5745B_MAPBGA256, MPC5745B_MAPBGA100
--	---

All of the above microcontroller devices are collectively named as MPC574XG .

2.2 Overview

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

AUTOSAR

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

2.3 About this Manual

This Technical Reference employs the following typographical conventions:

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

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

Notes and warnings are shown as below:

Note

This is a note.

2.4 Acronyms and Definitions

Table 2-2. Acronyms and Definitions

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
SUIL2	System Integration Unit Lite2
WKPU	Wakeup Unit
EMIOS	Enhanced Modular IO Subsystem

2.5 Reference List

Table 2-3. Reference List

#	Title	Version
1	AUTOSAR 4.2 Rev0002ICU Driver Software Specification Document.	4.2.2
2	MPC5748G Reference Manual	Rev. 5, 12/2016
3	MPC5746C Reference Manual	Rev. 4, 12/2016
4	MPC5748G_1N81M_Rev.2 (official document) (1N81M)	Jun-16
5	MPC5748G_1N81M_0N78S_Comparison_Summary_v 2_0 (internal document) (1N81M, 0N78S)	31.10.2016
6	MPC5746C_1N06M_Rev.4 (official document) (1N06M)	Jul-16

Table continues on the next page...

Table 2-3. Reference List (continued)

#	Title	Version
7	MPC5746C_cut1.1_cut2.0_cut2.1_comparison_v0 (internal document) (1N06M, 0N84S, 1N84S)	14-Sep-16
8	C3M_cut2.1_new_errata_20170113 (internal document) (1N84S)	13-Jan-17

Chapter 3 Driver

3.1 Requirements

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

3.2 Driver Design Summary

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

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

The eMIOS module of MPC574XG supports period time measurement, edge detection and notification, edge counting and edge time stamping.

The WKPU module of MPC574XG supports period time measurement, edge detection and notification, edge counting and edge time stamping.

The SIUL2 module of MPC574XG supports edge detection and notification. For details please refer to [Table 3-1](#).

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

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

For each user configured channel, a symbolic name is generated by the Tressos Studio configuration tool. The name shall be consequently used in upper applications (e.g. ICU_EMIO_0_CH_8).

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

3.3 Hardware Resources

The hardware configured by the Icu driver are SIUL2 and ETIMER.

3.4 Driver usage and configuration tips

None

3.5 Driver limitations

None

3.6 Hardware Channel Availability

Table 3-1. ICU Hardware channels availability for MPC574XG family

Device	Total eMIOS channels	Total External interrupt channels	Total WKPU interrupt channels
MPC5748G_MAPBGA324	96 ch, 16-bit	32 ch	32 ch
MPC5748G_MAPBGA256	96 ch, 16-bit	32 ch	32 ch
MPC5748G_LQFP176	96 ch, 16-bit	32 ch	32 ch

Table continues on the next page...

Table 3-1. ICU Hardware channels availability for MPC574XG family (continued)

MPC5748C_MAPBGA324	96 ch, 16-bit	32 ch	32 ch
MPC5748C_MAPBGA256	96 ch, 16-bit	32 ch	32 ch
MPC5748C_LQFP176	96 ch, 16-bit	32 ch	32 ch
MPC5747G_MAPBGA324	96 ch, 16-bit	32 ch	32 ch
MPC5747G_MAPBGA256	96 ch, 16-bit	32 ch	32 ch
MPC5747G_LQFP176	96 ch, 16-bit	32 ch	32 ch
MPC5747C_MAPBGA324	96 ch, 16-bit	32 ch	32 ch
MPC5747C_MAPBGA256	96 ch, 16-bit	32 ch	32 ch
MPC5747C_LQFP176	96 ch, 16-bit	32 ch	32 ch
MPC5746G_MAPBGA324	96 ch, 16-bit	32 ch	32 ch
MPC5746G_MAPBGA256	96 ch, 16-bit	32 ch	32 ch
MPC5746G_LQFP176	96 ch, 16-bit	32 ch	32 ch
MPC5746D_MAPBGA256	64 ch, 16-bit	25 ch	32 ch
MPC5746D_MAPBGA100	64 ch, 16-bit	25 ch	32 ch
MPC5746D_LQFP176	64 ch, 16-bit	25 ch	32 ch
MPC5746C_MAPBGA324	64 ch, 16-bit	25 ch	32 ch
MPC5746C_MAPBGA256	64 ch, 16-bit	25 ch	32 ch
MPC5746C_MAPBGA100	64 ch, 16-bit	25 ch	32 ch
MPC5746C_LQFP176	64 ch, 16-bit	25 ch	32 ch
MPC5746B_MAPBGA256	64 ch, 16-bit	25 ch	32 ch
MPC5746B_MAPBGA100	64 ch, 16-bit	25 ch	32 ch
MPC5746B_LQFP176	64 ch, 16-bit	25 ch	32 ch
MPC5745D_MAPBGA256	64 ch, 16-bit	25 ch	32 ch
MPC5745D_MAPBGA100	64 ch, 16-bit	25 ch	32 ch
MPC5745D_LQFP176	64 ch, 16-bit	25 ch	32 ch
MPC5745C_MAPBGA256	64 ch, 16-bit	25 ch	32 ch
MPC5745C_MAPBGA100	64 ch, 16-bit	25 ch	32 ch
MPC5745C_LQFP176	64 ch, 16-bit	25 ch	32 ch
MPC5745B_MAPBGA256	64 ch, 16-bit	25 ch	32 ch
MPC5745B_MAPBGA100	64 ch, 16-bit	25 ch	32 ch
MPC5745B_LQFP176	64 ch, 16-bit	25 ch	32 ch

3.7 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
ICU248	N/I	Icu_Lcfg.c shall include Icu_Cbk.h for a link time configuration if the call back function is linked to the module via the ROM structure.	Icu does not support link time. Icu has only precompile and post build variant (see ICU188 and ICU189)
ICU249	N/I	Icu_PBcfg.c shall include Icu_Cbk.h for post build time configuration if the call back function is linked to the module via the ROM structure.	Since the ICU is a driver module, it doesn't provide any callback functions for lower layer modules. (see chapter 8.4)
ICU250	N/I	Icu.c shall include Icu_Cbk.h for pre-compile time configuration	Since the ICU is a driver module, it doesn't provide any callback functions for lower layer modules. (see chapter 8.4)
ICU253	N/I	Icu_Lcfg.c shall include [ICU.h and MemMap.h.	Icu does not support link time. Icu has only precompile and post build variant (see ICU188 and ICU189)
ICU150	N/I	The Icu module shall not check the integrity if several calls for the same ICU channel are used during runtime in different tasks or ISRs.	The requirement is violating safety because ICU149 requirement is a safety integrity assumption for external environment, which shall be implemented for FTE; For GTE and NTE ICU149 has a role to increase availability because the check will be supported by ICU driver;
ICU350	N/I	Each variable that shall be accessible by AUTOSAR debugging, shall be defined as global variable.	ISO26262 Part6 -Table8- 1e: Avoid global variables or else justify their usage; Debugging concept not supported
ICU351	N/I	All type definitions of variables which shall be debugged shall be accessible by the header file Icu.h.	ISO26262 Part6 -Table8- 1e: Avoid global variables or else justify their usage; Debugging concept not supported
ICU352	N/I	The declaration of variables in the header file shall be such, that it is possible to calculate the size of the variables by C-'sizeof'.	Debugging concept not supported
ICU353	N/I	Variables available for debugging shall be described in the respective Basic Software Module description.	ISO26262 Part6 -Table8- 1e: Avoid global variables or else justify their usage; Debugging concept not supported
ICU039	N/F	Configuration of Signal Measurement - This container contains the configuration (parameters) in case the measurement mode is "ICU_MODE_SIGNAL_MEASUREMENT"	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

Table continues on the next page...

Table 3-3. ICU Deviations Table (continued)

Requirement	Status	Description	Notes
		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)	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
ICU131	N/S	The ICU driver shall describe which other modules (in which versions) are required. This description shall be done by the implementer.	Not an ICU driver requirement - this is a documentatio guideline
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
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)

3.8 eMIOS specific implementation details

The following sections contains the details of channels mode and master bus for each eMios channels.

On MPC574XG eMios channel support 3 modes: SAIC (Single Action InputCapture), IPW (Input Period Measurement) and IPWM (Input Pulse Width Measurement). Each eMios channel can use internal bus or external bus (Bus A, Bus F or Bus Diverse).

Below is the table depicts interaction between the channel mode and master bus can be used for each eMios channel ID (If there is an “X” in a table, it means "feature is supported" for that channel").

Table 3-4. Channels mode and Matser Bus details:

Channel ID	SAIC	IPW	IPWM	Bus A (Ch 23)	Bus F (CH 22)	Bus Diverse	Internal counter bus
EMIOS_0_CH_0	X			X	X	Bus B (Ch0)	X
EMIOS_0_CH_1	X	X	X	X	X	Bus B (Ch0)	X
EMIOS_0_CH_2	X	X	X	X	X	Bus B (Ch0)	X
EMIOS_0_CH_3	X	X	X	X	X	Bus B (Ch0)	X
EMIOS_0_CH_4	X	X	X	X	X	Bus B (Ch0)	X
EMIOS_0_CH_5	X	X	X	X	X	Bus B (Ch0)	X
EMIOS_0_CH_6	X	X	X	X	X	Bus B (Ch0)	X
EMIOS_0_CH_7	X	X	X	X	X	Bus B (Ch0)	X
EMIOS_0_CH_8	X			X	X	Bus C (Ch8)	X
EMIOS_0_CH_9	X	X	X	X	X	Bus C (Ch8)	
EMIOS_0_CH_10	X	X	X	X	X	Bus C (Ch8)	
EMIOS_0_CH_11	X	X	X	X	X	Bus C (Ch8)	
EMIOS_0_CH_12	X	X	X	X	X	Bus C (Ch8)	
EMIOS_0_CH_13	X	X	X	X	X	Bus C (Ch8)	
EMIOS_0_CH_14	X	X	X	X	X	Bus C (Ch8)	
EMIOS_0_CH_15	X	X	X	X	X	Bus C (Ch8)	
EMIOS_0_CH_16	X			X	X	Bus D (Ch16)	X
EMIOS_0_CH_17	X			X	X	Bus D (Ch16)	
EMIOS_0_CH_18	X			X	X	Bus D (Ch16)	
EMIOS_0_CH_19	X			X	X	Bus D (Ch16)	
EMIOS_0_CH_20	X			X	X	Bus D (Ch16)	
EMIOS_0_CH_21	X			X	X	Bus D (Ch16)	
EMIOS_0_CH_22	X			X	X	Bus D (Ch16)	X
EMIOS_0_CH_23	X			X	X	Bus D (Ch16)	X

Table continues on the next page...

Table 3-4. Channels mode and Matser Bus details: (continued)

Channel ID	SAIC	IPW	IPWM	Bus A (Ch 23)	Bus F (CH 22)	Bus Diverse	Internal counter bus
EMIOS_0_CH_24	X			X	X	Bus E (Ch24)	X
EMIOS_0_CH_25	X			X	X	Bus E (Ch24)	
EMIOS_0_CH_26	X			X	X	Bus E (Ch24)	
EMIOS_0_CH_27	X			X	X	Bus E (Ch24)	
EMIOS_0_CH_28	X			X	X	Bus E (Ch24)	
EMIOS_0_CH_29	X			X	X	Bus E (Ch24)	
EMIOS_0_CH_30	X			X	X	Bus E (Ch24)	
EMIOS_0_CH_31	X			X	X	Bus E (Ch24)	
EMIOS_1_CH_0	X			X	X	Bus B (Ch0)	X
EMIOS_1_CH_1	X	X	X	X	X	Bus B (Ch0)	
EMIOS_1_CH_2	X	X	X	X	X	Bus B (Ch0)	
EMIOS_1_CH_3	X	X	X	X	X	Bus B (Ch0)	
EMIOS_1_CH_4	X	X	X	X	X	Bus B (Ch0)	
EMIOS_1_CH_5	X	X	X	X	X	Bus B (Ch0)	
EMIOS_1_CH_6	X	X	X	X	X	Bus B (Ch0)	
EMIOS_1_CH_7	X	X	X	X	X	Bus B (Ch0)	
EMIOS_1_CH_8	X			X	X	Bus C (Ch8)	X
EMIOS_1_CH_9	X	X	X	X	X	Bus C (Ch8)	
EMIOS_1_CH_10	X	X	X	X	X	Bus C (Ch8)	
EMIOS_1_CH_11	X	X	X	X	X	Bus C (Ch8)	
EMIOS_1_CH_12	X	X	X	X	X	Bus C (Ch8)	
EMIOS_1_CH_13	X	X	X	X	X	Bus C (Ch8)	
EMIOS_1_CH_14	X	X	X	X	X	Bus C (Ch8)	
EMIOS_1_CH_15	X	X	X	X	X	Bus C (Ch8)	
EMIOS_1_CH_16	X			X	X	Bus D (Ch16)	X
EMIOS_1_CH_17	X			X	X	Bus D (Ch16)	
EMIOS_1_CH_18	X			X	X	Bus D (Ch16)	
EMIOS_1_CH_19	X			X	X	Bus D (Ch16)	
EMIOS_1_CH_20	X			X	X	Bus D (Ch16)	
EMIOS_1_CH_21	X			X	X	Bus D (Ch16)	
EMIOS_1_CH_22	X			X	X	Bus D (Ch16)	X
EMIOS_1_CH_23	X			X	X	Bus D (Ch16)	X
EMIOS_1_CH_24	X			X	X	Bus E (Ch24)	X
EMIOS_1_CH_25	X			X	X	Bus E (Ch24)	
EMIOS_1_CH_26	X			X	X	Bus E (Ch24)	
EMIOS_1_CH_27	X			X	X	Bus E (Ch24)	
EMIOS_1_CH_28	X			X	X	Bus E (Ch24)	
EMIOS_1_CH_29	X			X	X	Bus E (Ch24)	

Table continues on the next page...

Table 3-4. Channels mode and Matser Bus details: (continued)

Channel ID	SAIC	IPW	IPWM	Bus A (Ch 23)	Bus F (CH 22)	Bus Diverse	Internal counter bus
EMIOS_1_CH_30	X			X	X	Bus E (Ch24)	
EMIOS_1_CH_31	X			X	X	Bus E (Ch24)	
EMIOS_2_CH_0	X			X	X	Bus B (Ch0)	X
EMIOS_2_CH_1	X			X	X	Bus B (Ch0)	
EMIOS_2_CH_2	X			X	X	Bus B (Ch0)	
EMIOS_2_CH_3	X			X	X	Bus B (Ch0)	
EMIOS_2_CH_4	X			X	X	Bus B (Ch0)	
EMIOS_2_CH_5	X			X	X	Bus B (Ch0)	
EMIOS_2_CH_6	X			X	X	Bus B (Ch0)	
EMIOS_2_CH_7	X			X	X	Bus B (Ch0)	
EMIOS_2_CH_8	X			X	X	Bus C (Ch8)	X
EMIOS_2_CH_9	X			X	X	Bus C (Ch8)	
EMIOS_2_CH_10	X			X	X	Bus C (Ch8)	
EMIOS_2_CH_11	X			X	X	Bus C (Ch8)	
EMIOS_2_CH_12	X			X	X	Bus C (Ch8)	
EMIOS_2_CH_13	X			X	X	Bus C (Ch8)	
EMIOS_2_CH_14	X			X	X	Bus C (Ch8)	
EMIOS_2_CH_15	X			X	X	Bus C (Ch8)	
EMIOS_2_CH_16	X			X	X	Bus D (Ch16)	X
EMIOS_2_CH_17	X			X	X	Bus D (Ch16)	
EMIOS_2_CH_18	X			X	X	Bus D (Ch16)	
EMIOS_2_CH_19	X			X	X	Bus D (Ch16)	
EMIOS_2_CH_20	X			X	X	Bus D (Ch16)	
EMIOS_2_CH_21	X			X	X	Bus D (Ch16)	
EMIOS_2_CH_22	X			X	X	Bus D (Ch16)	X
EMIOS_2_CH_23	X			X	X	Bus D (Ch16)	X
EMIOS_2_CH_24	X			X	X	Bus E (Ch24)	X
EMIOS_2_CH_25	X			X	X	Bus E (Ch24)	
EMIOS_2_CH_26	X			X	X	Bus E (Ch24)	
EMIOS_2_CH_27	X			X	X	Bus E (Ch24)	
EMIOS_2_CH_28	X			X	X	Bus E (Ch24)	
EMIOS_2_CH_29	X			X	X	Bus E (Ch24)	
EMIOS_2_CH_30	X			X	X	Bus E (Ch24)	
EMIOS_2_CH_31	X			X	X	Bus E (Ch24)	

Note

Both ICU and PWM can be configured to use Master buses, but, due to the fact that the same Bus may not work with both modules, it is recommended that different busses should be used for each ASR module. This means that the user should be very careful when selecting what channel belongs to which module. For instance, if EMIOS0_CH1 is used by PWM and it configures the BUS B (CH 0) then the user SHOULD NOT configure any other ICU channel that uses BUS B as clock source and should select a channel that uses BUS C, D, E or INTERNAL_COUNTER.

3.9 Runtime Errors

This driver doesn't generate any runtime error.

3.10 Software specification

The following sections contains driver software specifications.

3.10.1 Define Reference

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

3.10.1.1 Define ICU_VENDOR_ID

Implements: Icu_interface

Table 3-5. Define ICU_VENDOR_ID Description

Name	ICU_VENDOR_ID
Initializer	43

3.10.1.2 Define ICU_AR_RELEASE_MAJOR_VERSION

Implements: Icu_interface

**Table 3-6. Define ICU_AR_RELEASE_MAJOR_VERSION
Description**

Name	ICU_AR_RELEASE_MAJOR_VERSION
Initializer	4

3.10.1.3 Define ICU_AR_RELEASE_MINOR_VERSION

Implements: Icu_interface

**Table 3-7. Define ICU_AR_RELEASE_MINOR_VERSION
Description**

Name	ICU_AR_RELEASE_MINOR_VERSION
Initializer	2

3.10.1.4 Define ICU_AR_RELEASE_REVISION_VERSION

Implements: Icu_interface

**Table 3-8. Define ICU_AR_RELEASE_REVISION_VERSION
Description**

Name	ICU_AR_RELEASE_REVISION_VERSION
Initializer	2

3.10.1.5 Define ICU_SW_MAJOR_VERSION

Implements: Icu_interface

Table 3-9. Define ICU_SW_MAJOR_VERSION
Description

Name	ICU_SW_MAJOR_VERSION
Initializer	1

3.10.1.6 Define ICU_SW_MINOR_VERSION

Implements: Icu_interface

Table 3-10. Define ICU_SW_MINOR_VERSION
Description

Name	ICU_SW_MINOR_VERSION
Initializer	0

3.10.1.7 Define ICU_SW_PATCH_VERSION

Implements: Icu_interface

Table 3-11. Define ICU_SW_PATCH_VERSION
Description

Name	ICU_SW_PATCH_VERSION
Initializer	0

3.10.1.8 Define ICU_CHECKWAKEUP_ID

API service ID for Icu_CheckWakeup function.

Details:

Parameters used when raising an error/exception

Table 3-12. Define ICU_CHECKWAKEUP_ID Description

Name	ICU_CHECKWAKEUP_ID
Initializer	(uint8)0x15

3.10.1.9 Define ICU_DEINIT_ID

API service ID for Icu_DeInit function.

Details:

Parameters used when raising an error/exception

Table 3-13. Define ICU_DEINIT_ID Description

Name	ICU_DEINIT_ID
Initializer	(uint8)0x01

3.10.1.10 Define ICU_DISABLEEDGECOUNT_ID

API service ID for Icu_DisableEdgeCount function.

Details:

Parameters used when raising an error/exception

Table 3-14. Define ICU_DISABLEEDGECOUNT_ID Description

Name	ICU_DISABLEEDGECOUNT_ID
Initializer	(uint8)0x0E

3.10.1.11 Define ICU_DISABLEEDGEDETECTION_ID

API service ID for Icu_DisableEdgeDetection function.

Details:

Parameters used when raising an error/exception

Table 3-15. Define ICU_DISABLEEDGEDETECTION_ID
Description

Name	ICU_DISABLEEDGEDETECTION_ID
Initializer	(uint8)0x17

3.10.1.12 Define ICU_DISABLENOTIFICATION_ID

API service ID for Icu_DisableNotification function.

Details:

Parameters used when raising an error/exception

Table 3-16. Define ICU_DISABLENOTIFICATION_ID
Description

Name	ICU_DISABLENOTIFICATION_ID
Initializer	(uint8)0x06

3.10.1.13 Define ICU_DISABLEWAKEUP_ID

API service ID for Icu_DisableWakeup function.

Details:

Parameters used when raising an error/exception

Table 3-17. Define ICU_DISABLEWAKEUP_ID
Description

Name	ICU_DISABLEWAKEUP_ID
Initializer	(uint8)0x03

3.10.1.14 Define ICU_ENABLEEDGECOUNT_ID

API service ID for Icu_EnableEdgeCount function.

Details:

Parameters used when raising an error/exception

Table 3-18. Define ICU_ENABLEEDGECOUNT_ID Description

Name	ICU_ENABLEEDGECOUNT_ID
Initializer	(uint8)0x0D

3.10.1.15 Define ICU_ENABLEEDGEDETECTION_ID

API service ID for Icu_EnableEdgeDetection function.

Details:

Parameters used when raising an error/exception

Table 3-19. Define ICU_ENABLEEDGEDETECTION_ID Description

Name	ICU_ENABLEEDGEDETECTION_ID
Initializer	(uint8)0x16

3.10.1.16 Define ICU_ENABLENOTIFICATION_ID

API service ID for Icu_EnableNotification function.

Details:

Parameters used when raising an error/exception

Table 3-20. Define ICU_ENABLENOTIFICATION_ID Description

Name	ICU_ENABLENOTIFICATION_ID
Initializer	(uint8)0x07

3.10.1.17 Define ICU_ENABLEWAKEUP_ID

API service ID for Icu_EnableWakeup function.

Details:

Parameters used when raising an error/exception

Table 3-21. Define ICU_ENABLEWAKEUP_ID Description

Name	ICU_ENABLEWAKEUP_ID
Initializer	(uint8)0x04

3.10.1.18 Define ICU_GET_INPUT_LEVEL_ID

API service ID for Icu_GetInputLevel function.

Details:

Parameters used when raising an error/exception

Table 3-22. Define ICU_GET_INPUT_LEVEL_ID Description

Name	ICU_GET_INPUT_LEVEL_ID
Initializer	(uint8)0x7CU

3.10.1.19 Define ICU_GETDUTYCYCLEVALUES_ID

API service ID for Icu_GetDutyCycleValues function.

Details:

Parameters used when raising an error/exception

Table 3-23. Define ICU_GETDUTYCYCLEVALUES_ID Description

Name	ICU_GETDUTYCYCLEVALUES_ID
Initializer	(uint8)0x11

3.10.1.20 Define ICU_GETEDGENUMBERS_ID

API service ID for Icu_GetEdgeNumbers function.

Details:

Parameters used when raising an error/exception

Table 3-24. Define ICU_GETEDGENUMBERS_ID Description

Name	ICU_GETEDGENUMBERS_ID
Initializer	(uint8)0x0F

3.10.1.21 Define ICU_GETINPUTSTATE_ID

API service ID for Icu_GetInputState function.

Details:

Parameters used when raising an error/exception

Table 3-25. Define ICU_GETINPUTSTATE_ID Description

Name	ICU_GETINPUTSTATE_ID
Initializer	(uint8)0x08

3.10.1.22 Define ICU_GETTIMEELAPSED_ID

API service ID for Icu_GetTimeElapsed function.

Details:

Parameters used when raising an error/exception

Table 3-26. Define ICU_GETTIMEELAPSED_ID Description

Name	ICU_GETTIMEELAPSED_ID
-------------	-----------------------

Table continues on the next page...

**Table 3-26. Define ICU_GETTIMEELAPSED_ID Description
(continued)**

Initializer	(uint8)0x10
--------------------	-------------

3.10.1.23 Define ICU_GETTIMESTAMPINDEX_ID

API service ID for Icu_GetTimestampIndex function.

Details:

Parameters used when raising an error/exception

Table 3-27. Define ICU_GETTIMESTAMPINDEX_ID Description

Name	ICU_GETTIMESTAMPINDEX_ID
Initializer	(uint8)0x0B

3.10.1.24 Define ICU_GETVERSIONINFO_ID

API service ID for Icu_GetVersionInfo function.

Details:

Parameters used when raising an error/exception

Table 3-28. Define ICU_GETVERSIONINFO_ID Description

Name	ICU_GETVERSIONINFO_ID
Initializer	(uint8)0x12

3.10.1.25 Define ICU_INIT_ID

API service ID for Icu_Init function.

Details:

Parameters used when raising an error/exception

Table 3-29. Define ICU_INIT_ID Description

Name	ICU_INIT_ID
Initializer	(uint8)0x00

3.10.1.26 Define ICU_MODULE_ID

Implements: Icu_interface

Table 3-30. Define ICU_MODULE_ID Description

Name	ICU_MODULE_ID
Initializer	122

3.10.1.27 Define ICU_RESETEGECOUNT_ID

API service ID for Icu_ResetEdgeCount function.

Details:

Parameters used when raising an error/exception

Table 3-31. Define ICU_RESETEGECOUNT_ID Description

Name	ICU_RESETEGECOUNT_ID
Initializer	(uint8)0x0C

3.10.1.28 Define ICU_SETACTIVATIONCONDITION_ID

API service ID for Icu_SetActivationCondition function.

Details:

Parameters used when raising an error/exception

Table 3-32. Define ICU_SETACTIVATIONCONDITION_ID Description

Name	ICU_SETACTIVATIONCONDITION_ID
Initializer	(uint8)0x05

3.10.1.29 Define ICU_SETMODE_ID

API service ID for Icu_SetMode function.

Details:

Parameters used when raising an error/exception

Table 3-33. Define ICU_SETMODE_ID Description

Name	ICU_SETMODE_ID
Initializer	(uint8)0x02

3.10.1.30 Define ICU_STARTSIGNALMEASUREMENT_ID

API service ID for Icu_StartSignalMeasurement function.

Details:

Parameters used when raising an error/exception

Table 3-34. Define ICU_STARTSIGNALMEASUREMENT_ID Description

Name	ICU_STARTSIGNALMEASUREMENT_ID
Initializer	(uint8)0x13

3.10.1.31 Define ICU_STARTTIMESTAMP_ID

API service ID for Icu_StartTimestamp function.

Details:

Parameters used when raising an error/exception

Table 3-35. Define ICU_STARTTIMESTAMP_ID Description

Name	ICU_STARTTIMESTAMP_ID
Initializer	(uint8)0x09

3.10.1.32 Define ICU_STOPSIGNALMEASUREMENT_ID

API service ID for Icu_StopSignalMeasurement function.

Details:

Parameters used when raising an error/exception

Table 3-36. Define ICU_STOPSIGNALMEASUREMENT_ID Description

Name	ICU_STOPSIGNALMEASUREMENT_ID
Initializer	(uint8)0x14

3.10.1.33 Define ICU_STOPTIMESTAMP_ID

API service ID for Icu_StopTimestamp function.

Details:

Parameters used when raising an error/exception

Table 3-37. Define ICU_STOPTIMESTAMP_ID Description

Name	ICU_STOPTIMESTAMP_ID
Initializer	(uint8)0x0A

3.10.1.34 Define ICU_SET_CLOCK_MODE_ID

API service ID for Icu_SetClockMode function.

Details:

Parameters used when raising an error/exception

Table 3-38. Define ICU_SET_CLOCK_MODE_ID Description

Name	ICU_SET_CLOCK_MODE_ID
Initializer	(uint8)0x7B

3.10.1.35 Define ICU_GET_PULSE_WIDTH_ID

API service ID for Icu_GetPulseWidth function.

Details:

Parameters used when raising an error/exception

Table 3-39. Define ICU_GET_PULSE_WIDTH_ID Description

Name	ICU_GET_PULSE_WIDTH_ID
Initializer	(uint8)0x1AU

3.10.1.36 Define ICU_GET_CAPTURE_REGISTER_VALUE_ID

API service ID for Icu_StartSignalMeasurement function.

Details:

Parameters used when raising an error/exception

Table 3-40. Define ICU_GET_CAPTURE_REGISTER_VALUE_ID Description

Name	ICU_GET_CAPTURE_REGISTER_VALUE_ID
Initializer	(uint8)0x19U

3.10.1.37 Define ICU_E_ALREADY_INITIALIZED

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

Implements: Icu_ErrorCodes_define

Table 3-41. Define ICU_E_ALREADY_INITIALIZED
Description

Name	ICU_E_ALREADY_INITIALIZED
Initializer	(uint8)0x17

3.10.1.38 Define ICU_E_BUSY_OPERATION

API service Icu_SetMode is called while a running operation.

Implements: Icu_ErrorCodes_define

Table 3-42. Define ICU_E_BUSY_OPERATION
Description

Name	ICU_E_BUSY_OPERATION
Initializer	(uint8)0x16

3.10.1.39 Define ICU_E_EDGECOUNTING_OVERFLOW

API Icu_GetEdgeNumbers service called when the Counter rolls over.

Implements: Icu_ErrorCodes_define

Table 3-43. Define ICU_E_EDGECOUNTING_OVERFLOW
Description

Name	ICU_E_EDGECOUNTING_OVERFLOW
Initializer	(uint8)0xD0

3.10.1.40 Define ICU_E_MEASUREMENT_OVERFLOW

API Icu_GetTimeElapsed service called when the Time elapsed overflows.

Implements: Icu_ErrorCodes_define

**Table 3-44. Define ICU_E_MEASUREMENT_OVERFLOW
Description**

Name	ICU_E_MEASUREMENT_OVERFLOW
Initializer	(uint8)0xD2

3.10.1.41 Define ICU_E_NOT_STARTED

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

Implements: Icu_ErrorCodes_define

Table 3-45. Define ICU_E_NOT_STARTED Description

Name	ICU_E_NOT_STARTED
Initializer	(uint8)0x15

3.10.1.42 Define ICU_E_PARAM_ACTIVATION

API service used with an invalid or not feasible activation.

Implements: Icu_ErrorCodes_define

Table 3-46. Define ICU_E_PARAM_ACTIVATION Description

Name	ICU_E_PARAM_ACTIVATION
Initializer	(uint8)0x0C

3.10.1.43 Define ICU_E_PARAM_POINTER

API service used with an invalid application-buffer pointer.

Implements: Icu_ErrorCodes_define

Table 3-47. Define ICU_E_PARAM_POINTER Description

Name	ICU_E_PARAM_POINTER
Initializer	(uint8)0x0A

3.10.1.44 Define ICU_E_PARAM_BUFFER_SIZE

API service used with an invalid buffer size.

Implements: Icu_ErrorCodes_define

Table 3-48. Define ICU_E_PARAM_BUFFER_SIZE Description

Name	ICU_E_PARAM_BUFFER_SIZE
Initializer	(uint8)0x0E

3.10.1.45 Define ICU_E_PARAM_CHANNEL

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

Implements: Icu_ErrorCodes_define

Table 3-49. Define ICU_E_PARAM_CHANNEL Description

Name	ICU_E_PARAM_CHANNEL
Initializer	(uint8)0x0B

3.10.1.46 Define ICU_E_PARAM_MODE

API service Icu_SetMode used with an invalid mode.

Implements: Icu_ErrorCodes_define

Table 3-50. Define ICU_E_PARAM_MODE Description

Name	ICU_E_PARAM_MODE
Initializer	(uint8)0x0F

3.10.1.47 Define ICU_E_PARAM_NOTIFY_INTERVAL

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

Implements: Icu_ErrorCodes_define

Table 3-51. Define ICU_E_PARAM_NOTIFY_INTERVAL Description

Name	ICU_E_PARAM_NOTIFY_INTERVAL
Initializer	(uint8)0x18

3.10.1.48 Define ICU_E_PARAM_VINFO

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

Implements: Icu_ErrorCodes_define

Table 3-52. Define ICU_E_PARAM_VINFO Description

Name	ICU_E_PARAM_VINFO
Initializer	(uint8)0x19

3.10.1.49 Define ICU_E_TIMESTAMP_OVERFLOW

API Icu_GetTimestampIndex service called when the Time stamp count overflows.

Implements: Icu_ErrorCodes_define

**Table 3-53. Define ICU_E_TIMESTAMP_OVERFLOW
Description**

Name	ICU_E_TIMESTAMP_OVERFLOW
Initializer	(uint8)0xD1

3.10.1.50 Define ICU_E_UNINIT

API service used without module initialization.

Implements: Icu_ErrorCodes_define

Table 3-54. Define ICU_E_UNINIT Description

Name	ICU_E_UNINIT
Initializer	(uint8)0x14

3.10.1.51 Define ICU_E_INVALID_MODE

API Icu_EnableWakeup and Icu_DisableWakeup services called in ICU_SLEEP_MODE.

Implements: Icu_ErrorCodes_define

Table 3-55. Define ICU_E_INVALID_MODE Description

Name	ICU_E_INVALID_MODE
Initializer	(uint8)0xD3

3.10.1.52 Define ICU_E_FORBIDEN_MODE

API Check mode before set up feature for channel.

Implements: Icu_ErrorCodes_define

Table 3-56. Define ICU_E_FORBIDEN_MODE Description

Name	ICU_E_FORBIDEN_MODE
Initializer	(uint8)0xD4

3.10.1.53 Define ICU_E_PARAM_CLOCK_MODE

API Icu_GetTimestampIndex service called when the Time stamp count overflows.

Implements: Icu_ErrorCodes_define

Table 3-57. Define ICU_E_PARAM_CLOCK_MODE Description

Name	ICU_E_PARAM_CLOCK_MODE
Initializer	(uint8)0x7A

3.10.1.54 Define ICU_E_INIT_FAILED

API Icu_Init service called with wrong parameter.

Implements: Icu_ErrorCodes_define

Table 3-58. Define ICU_E_INIT_FAILED Description

Name	ICU_E_INIT_FAILED
Initializer	(uint8)0x0D

3.10.1.55 Define ICU_GET_VERSION_INFO_API

Configuration of Optional API's.

Details:

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

STD_ON: Icu_GetVersionInfo() can be used. STD_OFF: Icu_GetVersionInfo() can not be used.

Table 3-59. Define ICU_GET_VERSION_INFO_API Description

Name	ICU_GET_VERSION_INFO_API
Initializer	(STD_ON)

3.10.1.56 Define ICU_DE_INIT_API

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

Implements: ICU_DE_INIT_API_define

Table 3-60. Define ICU_DE_INIT_API Description

Name	ICU_DE_INIT_API
Initializer	(STD_ON)

3.10.1.57 Define ICU_SET_MODE_API

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

Implements: ICU_SET_MODE_API_define

Table 3-61. Define ICU_SET_MODE_API Description

Name	ICU_SET_MODE_API
Initializer	(STD_ON)

3.10.1.58 Define ICU_DISABLE_WAKEUP_API

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

Implements: ICU_DISABLE_WAKEUP_API_define

Table 3-62. Define ICU_DISABLE_WAKEUP_API Description

Name	ICU_DISABLE_WAKEUP_API
Initializer	(STD_ON)

3.10.1.59 Define ICU_ENABLE_WAKEUP_API

Adds or removes the service `Icu_EnableWakeup()` from the code.

STD_ON: `Icu_EnableWakeup()` can be used. STD_OFF: `Icu_EnableWakeup()` can not be used.

Implements: ICU_ENABLE_WAKEUP_API_define

Table 3-63. Define ICU_ENABLE_WAKEUP_API Description

Name	ICU_ENABLE_WAKEUP_API
Initializer	(STD_ON)

3.10.1.60 Define ICU_TIMESTAMP_API

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

Implements: ICU_TIMESTAMP_API_define

Table 3-64. Define ICU_TIMESTAMP_API Description

Name	ICU_TIMESTAMP_API
Initializer	(STD_ON)

3.10.1.61 Define ICU_EDGE_COUNT_API

Adds or removes all services related to the edge counting functionality as listed below, from the

code: `Icu_ResetEdgeCount()`, `Icu_EnableEdgeCount()`, `Icu_DisableEdgeCount()`, `Icu_GetEdgeNumbers()`.

STD_ON: The services listed above can be used. STD_OFF: The services listed above can not be used. .

Table 3-65. Define ICU_EDGE_COUNT_API Description

Name	ICU_EDGE_COUNT_API
Initializer	(STD_ON)

3.10.1.62 Define ICU_GET_TIME_ELAPSED_API

Adds or removes the service `Icu_GetTimeElapsed()` from the code.

STD_ON: `Icu_GetTimeElapsed()` can be used. STD_OFF: `Icu_GetTimeElapsed()` can not be used.

Implements: ICU_GET_TIME_ELAPSED_API_define

Table 3-66. Define ICU_GET_TIME_ELAPSED_API Description

Name	ICU_GET_TIME_ELAPSED_API
Initializer	(STD_ON)

3.10.1.63 Define ICU_GET_DUTY_CYCLE_VALUES_API

Adds or removes the service `Icu_GetDutyCycleValues()` from the code.

STD_ON: `Icu_GetDutyCycleValues()` can be used. STD_OFF: `Icu_GetDutyCycleValues()` can not be used.

Implements: ICU_GET_DUTY_CYCLE_VALUES_API_define

Table 3-67. Define ICU_GET_DUTY_CYCLE_VALUES_API Description

Name	ICU_GET_DUTY_CYCLE_VALUES_API
Initializer	(STD_ON)

3.10.1.64 Define ICU_GET_INPUT_STATE_API

Adds or removes the service `Icu_GetInputState()` from the code.

STD_ON:`Icu_GetInputState()` can be used. STD_OFF:`Icu_GetInputState()` can not be used.

Implements: ICU_GET_INPUT_STATE_API_define

Table 3-68. Define ICU_GET_INPUT_STATE_API Description

Name	ICU_GET_INPUT_STATE_API
Initializer	(STD_ON)

3.10.1.65 Define ICU_SIGNAL_MEASUREMENT_API

Adds or removes the services `Icu_StartSignalMeasurement()` and

`Icu_StopSignalMeasurement()` from the code. STD_ON:`Icu_StartSignalMeasurement()` and

`Icu_StopSignalMeasurement()` can be used. STD_OFF:`Icu_StartSignalMeasurement()` and

`Icu_StopSignalMeasurement()` can not be used.

Implements: ICU_SIGNAL_MEASUREMENT_API_define

Table 3-69. Define ICU_SIGNAL_MEASUREMENT_API Description

Name	ICU_SIGNAL_MEASUREMENT_API
Initializer	(STD_ON)

3.10.1.66 Define ICU_WAKEUP_FUNCTIONALITY_API

Adds or removes the service `Icu_CheckWakeup()` from the code.

STD_ON:`Icu_CheckWakeup()` can be used. STD_OFF:`Icu_CheckWakeup()` can not be used.

Implements: ICU_WAKEUP_FUNCTIONALITY_API_define

Table 3-70. Define ICU_WAKEUP_FUNCTIONALITY_API Description

Name	ICU_WAKEUP_FUNCTIONALITY_API
Initializer	(STD_ON)

3.10.1.67 Define ICU_EDGE_DETECT_API

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

Implements: ICU_EDGE_DETECT_API_define

Table 3-71. Define ICU_EDGE_DETECT_API Description

Name	ICU_EDGE_DETECT_API
Initializer	(STD_ON)

3.10.1.68 Define ICU_OVERFLOW_NOTIFICATION_API

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

Table 3-72. Define ICU_OVERFLOW_NOTIFICATION_API Description

Name	ICU_OVERFLOW_NOTIFICATION_API
Initializer	(STD_ON)

3.10.1.69 Define ICU_GET_INPUT_LEVEL_API

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

Table 3-73. Define ICU_GET_INPUT_LEVEL_API
Description

Name	ICU_GET_INPUT_LEVEL_API
Initializer	(STD_ON)

3.10.1.70 Define ICU_DUAL_CLOCK_MODE_API

Adds or removes the service `Icu_SetClockMode()` from the code.

STD_ON: `Icu_SetClockMode()` can be used. STD_OFF: `Icu_SetClockMode()` can not be used.

Implements: ICU_DUAL_CLOCK_MODE_API_define

Table 3-74. Define ICU_DUAL_CLOCK_MODE_API
Description

Name	ICU_DUAL_CLOCK_MODE_API
Initializer	(STD_ON)

3.10.1.71 Define ICU_TIMESTAMP_USES_DMA

Implements: ICU_TIMESTAMP_USES_DMA_define

Table 3-75. Define ICU_TIMESTAMP_USES_DMA
Description

Name	ICU_TIMESTAMP_USES_DMA
Initializer	(STD_ON)

3.10.1.72 Define ICU_SIGNALMEASUREMENT_USES_DMA

Implements: ICU_SIGNALMEASUREMENT_USES_DMA_define

Table 3-76. Define ICU_SIGNALMEASUREMENT_USES_DMA
Description

Name	ICU_SIGNALMEASUREMENT_USES_DMA
Initializer	(STD_ON)

3.10.1.73 Define ICU_GET_PULSE_WIDTH_API

Adds or removes the service `Icu_GetPulseWidth()` from the code.

STD_ON: `Icu_GetPulseWidth()` can be used. STD_OFF: `Icu_GetPulseWidth()` can not be used.

Implements: ICU_GET_PULSE_WIDTH_API_define

Table 3-77. Define ICU_GET_PULSE_WIDTH_API
Description

Name	ICU_GET_PULSE_WIDTH_API
Initializer	(STD_ON)

3.10.1.74 Define ICU_CAPTUREREGISTER_API

Adds or removes the service `Icu_GetCaptureRegisterValue()` from the code.

STD_ON: `Icu_GetCaptureRegisterValue()` can be used.

STD_OFF: `Icu_GetCaptureRegisterValue()` can not be used.

Implements: ICU_CAPTUREREGISTER_API_define

Table 3-78. Define ICU_CAPTUREREGISTER_API
Description

Name	ICU_CAPTUREREGISTER_API
Initializer	(STD_ON)

3.10.1.75 Define ICU_REPORT_WAKEUP_SOURCE

Switch for enabling Wakeup source reporting. STD_ON: Report Wakeup source.

STD_OFF: Do not report Wakeup source.

Table 3-79. Define ICU_REPORT_WAKEUP_SOURCE Description

Name	ICU_REPORT_WAKEUP_SOURCE
Initializer	(STD_ON)

3.10.1.76 Define ICU_DEV_ERROR_DETECT

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

Implements: ICU_DEV_ERROR_DETECT_define

Table 3-80. Define ICU_DEV_ERROR_DETECT Description

Name	ICU_DEV_ERROR_DETECT
Initializer	(STD_ON)

3.10.2 Enum Reference

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

3.10.2.1 Enumeration Icu_ActivationType

Definition of the type of activation of an ICU channel.

Implements: Icu_ActivationType_enumeration

Table 3-81. Enumeration Icu_ActivationType Values

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

3.10.2.2 Enumeration Icu_InputStateType

Input state of an ICU channel.

Implements: Icu_InputState_enumeration

Table 3-82. Enumeration Icu_InputStateType Values

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

3.10.2.3 Enumeration Icu_MeasurementModeType

Definition of the measurement mode type.

Implements: Icu_MeasurementModeType_enumeration

Table 3-83. Enumeration Icu_MeasurementModeType Values

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

3.10.2.4 Enumeration Icu_ModeType

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

Implements: Icu_ModeType_enumeration

Table 3-84. Enumeration Icu_ModeType Values

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

3.10.2.5 Enumeration Icu_SignalMeasurementPropertyType

Definition of the measurement property type.

Implements: Icu_SignalMeasurementPropertyType_enumeration

Table 3-85. Enumeration Icu_SignalMeasurementPropertyType Values

Name	Initializer	Description
ICU_LOW_TIME	0x01U	The channel is configured for reading the elapsed Signal Low Time
ICU_HIGH_TIME	0x02U	The channel is configured for reading the elapsed Signal High Time
ICU_PERIOD_TIME	0x04U	The channel is configured for reading the elapsed Signal Period Time
ICU_DUTY_CYCLE	0x07U	The channel is configured to read values which are needed for calculating the duty cycle (coherent Active and Period Time).

3.10.2.6 Enumeration Icu_TimestampBufferType

Definition of the timestamp measurement property type.

Implements: Icu_TimestampBufferType_enumeration

Table 3-86. Enumeration Icu_TimestampBufferType Values

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

3.10.2.7 Enumeration Icu_SelectPrescalerType

Definition of the clock prescaler type.

Implements: Icu_SelectPrescalerType_enumeration

Table 3-87. Enumeration Icu_SelectPrescalerType Values

Name	Initializer	Description
ICU_NORMAL_CLOCK_MODE	0U	The default prescaler used in the driver
ICU_ALTERNATE_CLOCK_MODE	1U	The alternate prescaler that can be used for user defined applications (e.g. low power mode).

3.10.2.8 Enumeration Icu_LevelType

Return the status of the pin. Enumeration of to check the status of emios pin.

Implements: Icu_LevelType_enumeration

Table 3-88. Enumeration Icu_LevelType Values

Name	Initializer	Description
ICU_LEVEL_LOW	0U	Default Input PIN Status.
ICU_LEVEL_HIGH	1U	As Icu_GetInputState do not give the Actual PIN status user can call the Non Autosar API Icu_GetInputLevel to get the Actual status of PIN.

3.10.2.9 Enumeration Icu_Siul2_ActivationType

Definition of the type of activation of an Siul2 channel.

Implements: Icu_Siul2_ActivationType_enumeration

Table 3-89. Enumeration Icu_Siul2_ActivationType Values

Name	Initializer	Description
SIUL2_FALLING_EDGE	0x0U	An appropriate action shall be executed when a falling edge occurs on the ICU input signal.
SIUL2_RISING_EDGE	0x1U	An appropriate action shall be executed when a rising edge occurs on the ICU input signal.
SIUL2_BOTH_EDGES	0x2U	An appropriate action shall be executed when a falling or rising edge occurs on the ICU input signal.

3.10.2.10 Enumeration Icu_Siul2_SelectPrescalerType

Definition of the clock prescaler type.

Implements: Icu_Siul2_SelectPrescalerType_enumeration

Table 3-90. Enumeration Icu_Siul2_SelectPrescalerType Values

Name	Initializer	Description
SIUL2_ICU_NORMAL_CLOCK_MODE	0U	The default prescaler used in the driver
SIUL2_ICU_ALTERNATE_CLOCK_MODE	1U	The alternate prescaler that can be used for user defined applications (e.g. low power mode).

3.10.3 Function Reference

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

3.10.3.1 Function Icu_Init

This function initializes the driver.

Details:

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

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

Return: void.

Implements: Icu_Init_Activity

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

Prototype: void Icu_Init(const Icu_ConfigType *ConfigPtr);

Table 3-91. Icu_Init Arguments

Type	Name	Direction	Description
constIcu_ConfigType*	ConfigPtr	input	Pointer to a selected configuration structure.

3.10.3.2 Function Icu_DeInit

This function de-initializes the ICU module.

Details:

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

Return: void.

Pre: Icu_Init must be called before.

Implements: Icu_DeInit_Activity

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

Prototype: void Icu_DeInit(void);

3.10.3.3 Function Icu_SetMode

This function sets the ICU mode.

Details:

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

Return: void.

Pre: Icu_Init must be called before.

Implements: Icu_SetMode_Activity

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

Prototype: void Icu_SetMode(Icu_ModeType Mode);

Table 3-92. Icu_SetMode Arguments

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

3.10.3.4 Function Icu_GetVersionInfo

This service returns the version information of this module.

Details:

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

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

Return: void.

Implements: Icu_GetVersionInfo_Activity

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

Prototype: void Icu_GetVersionInfo(Std_VersionInfoType *versioninfo);

Table 3-93. Icu_GetVersionInfo Arguments

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

3.10.3.5 Function Icu_CheckWakeup

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

Details:

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

Return: void.

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

Implements: Icu_CheckWakeup_Activity

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

Prototype: void Icu_CheckWakeup(EcuM_WakeupSourceType WakeupSource);

Table 3-94. Icu_CheckWakeup Arguments

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

3.10.3.6 Function Icu_DisableWakeup

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

Details:

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

Return: void.

Pre: Icu_Init must be called before.

Implements: Icu_DisableWakeup_Activity

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

Prototype: void Icu_DisableWakeup(Icu_ChannelType Channel);

Table 3-95. Icu_DisableWakeup Arguments

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

3.10.3.7 Function Icu_EnableWakeup

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

Details:

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

Return: void.

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

Implements: Icu_EnableWakeup_Activity

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

Prototype: void Icu_EnableWakeup(Icu_ChannelType Channel);

Table 3-96. Icu_EnableWakeup Arguments

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

3.10.3.8 Function Icu_DisableEdgeCount

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

Details:

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

Return: void.

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

Implements: Icu_DisableEdgeCount_Activity

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

Prototype: void Icu_DisableEdgeCount(Icu_ChannelType Channel);

Table 3-97. Icu_DisableEdgeCount Arguments

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

3.10.3.9 Function Icu_EnableEdgeCount

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

Details:

This service is reentrant and shall enable the counting of edges of the given channel. Note: This service does not do the real counting itself. This is done by the hardware (capture unit). Only the configured edges shall be counted (rising edge, falling edge or both edges).

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

Return: void.

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

Implements: Icu_EnableEdgeCount_Activity

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

Prototype: void Icu_EnableEdgeCount(Icu_ChannelType Channel);

Table 3-98. Icu_EnableEdgeCount Arguments

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

3.10.3.10 Function Icu_ResetEdgeCount

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

Details:

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

Return: void.

Pre: Icu_Init must be called before.

Implements: Icu_ResetEdgeCount_Activity

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

Prototype: void Icu_ResetEdgeCount(Icu_ChannelType Channel);

Table 3-99. Icu_ResetEdgeCount Arguments

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

3.10.3.11 Function Icu_GetEdgeNumbers

This function reads the number of counted edges.

Details:

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

Return: `Icu_EdgeNumberType` - Number of the counted edges.

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

Implements: `Icu_GetEdgeNumbers_Activity`

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

Prototype: `Icu_EdgeNumberType Icu_GetEdgeNumbers(Icu_ChannelType Channel);`

Table 3-100. Icu_GetEdgeNumbers Arguments

Type	Name	Direction	Description
<code>Icu_ChannelType</code>	Channel	input	Numeric identifier of the ICU channel.

3.10.3.12 Function `Icu_DisableEdgeDetection`

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

Details:

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

Return: `void`.

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

Implements: `Icu_DisableEdgeDetection_Activity`

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

Prototype: `void Icu_DisableEdgeDetection(Icu_ChannelType Channel);`

Table 3-101. Icu_DisableEdgeDetection Arguments

Type	Name	Direction	Description
<code>Icu_ChannelType</code>	Channel	input	Numeric identifier of the ICU channel.

3.10.3.13 Function Icu_EnableEdgeDetection

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

Details:

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

Return: void.

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

Implements: Icu_EnableEdgeDetection_Activity

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

Prototype: void Icu_EnableEdgeDetection(Icu_ChannelType Channel);

Table 3-102. Icu_EnableEdgeDetection Arguments

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

3.10.3.14 Function Icu_DisableNotification

This function disables the notification of a channel.

Details:

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

Return: void.

Pre: Icu_Init must be called before.

Implements: Icu_DisableNotification_Activity

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

Prototype: void Icu_DisableNotification(Icu_ChannelType Channel);

Table 3-103. Icu_DisableNotification Arguments

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

3.10.3.15 Function Icu_EnableNotification

This function enables the notification on the given channel.

Details:

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

Return: void.

Pre: Icu_Init must be called before.

Implements: Icu_EnableNotification_Activity

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

Prototype: void Icu_EnableNotification(Icu_ChannelType Channel);

Table 3-104. Icu_EnableNotification Arguments

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

3.10.3.16 Function Icu_SetActivationCondition

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

Details:

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

- ICU_MODE_SIGNAL_EDGE_DETECT

- ICU_MODE_TIMESTAMP
- ICU_MODE_EDGE_COUNTER

Return: void.

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

Implements: Icu_SetActivationCondition_Activity

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

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

Table 3-105. Icu_SetActivationCondition Arguments

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

3.10.3.17 Function Icu_GetInputLevel

This function returns the actual status of PIN.

Details:

This function returns the actual status of PIN.

Return: Icu_LevelType.

Pre: Icu_Init must be called before.

Implements: Icu_GetInputLevel_Activity

Prototype: Icu_LevelType Icu_GetInputLevel(Icu_ChannelType Channel);

Table 3-106. Icu_GetInputLevel Arguments

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

3.10.3.18 Function Icu_GetInputState

This function returns the status of the ICU input.

Details:

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

- ICU_MODE_SIGNAL_EDGE_DETECT,
- ICU_MODE_SIGNAL_MEASUREMENT.

Return: Icu_InputStateType.

Pre: Icu_Init must be called before.

Implements: Icu_GetInputState_Activity

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

Prototype: Icu_InputStateType Icu_GetInputState(Icu_ChannelType Channel);

Table 3-107. Icu_GetInputState Arguments

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

Table 3-108. Icu_GetInputState Return Values

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

3.10.3.19 Function Icu_StartSignalMeasurement

This function starts the measurement of signals.

Details:

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

Return: void.

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

Implements: Icu_StartSignalMeasurement_Activity

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

Prototype: void Icu_StartSignalMeasurement(Icu_ChannelType Channel);

Table 3-109. Icu_StartSignalMeasurement Arguments

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

3.10.3.20 Function Icu_StopSignalMeasurement

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

Details:

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

Return: void.

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

Implements: Icu_StopSignalMeasurement_Activity

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

Prototype: void Icu_StopSignalMeasurement(Icu_ChannelType Channel);

Table 3-110. Icu_StopSignalMeasurement Arguments

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

3.10.3.21 Function Icu_GetDutyCycleValues

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

Details:

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

Return: void.

@ implements Icu_GetDutyCycleValues_Activity

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

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

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

Table 3-111. Icu_GetDutyCycleValues Arguments

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

3.10.3.22 Function Icu_GetTimeElapsed

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

Details:

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

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

configurable.

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

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

Implements: Icu_GetTimeElapsed_Activity

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

Prototype: Icu_ValueType Icu_GetTimeElapsed(Icu_ChannelType Channel);

Table 3-112. Icu_GetTimeElapsed Arguments

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

3.10.3.23 Function Icu_GetTimestampIndex

This function reads the timestamp index of the given channel.

Details:

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

Return: Icu_IndexType - Timestamp index of the given channel.

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

Implements: Icu_GetTimestampIndex_Activity

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

Prototype: Icu_IndexType Icu_GetTimestampIndex(Icu_ChannelType Channel);

Table 3-113. Icu_GetTimestampIndex Arguments

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

3.10.3.24 Function Icu_StartTimestamp

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

Details:

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

Return: void.

Pre: Icu_Init must be called before.

Implements: Icu_StartTimestamp_Activity

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

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

Table 3-114. Icu_StartTimestamp Arguments

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

3.10.3.25 Function Icu_StopTimestamp

This function stops the timestamp measurement of the given channel.

Details:

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

Return: void.

Pre: Icu_Init must be called before.

Implements: Icu_StopTimestamp_Activity

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

Prototype: void Icu_StopTimestamp(Icu_ChannelType Channel);

Table 3-115. Icu_StopTimestamp Arguments

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

3.10.3.26 Function Icu_SetClockMode

This function sets the clock mode for the driver

Details:

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

Return: void.

Pre: Icu_Init must be called before.

Implements: Icu_SetClockMode_Activity

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

Prototype: void Icu_SetClockMode(Icu_SelectPrescalerType Prescaler);

Table 3-116. Icu_SetClockMode Arguments

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

3.10.3.27 Function Icu_GetCaptureRegisterValue

This service returns the value of Capture register.

Details:

This service returns the value of Capture register.

Return: Icu_ValueType.

Pre: Icu_Init must be called before.

Implements: Icu_GetCaptureRegisterValue_Activity

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

Prototype: Icu_ValueType Icu_GetCaptureRegisterValue_Activity(Icu_ChannelType Channel);

Table 3-117. Icu_GetCaptureRegisterValue_Activity Arguments

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

3.10.3.28 Function Icu_GetPulseWidth

This function starts the measurement of signals.

Details:

This function returns the signals High time, Low time and Period without using the channel interrupt.

Return: void.

Pre: Icu_Init must be called before.

Implements: Icu_GetPulseWidth_Activity

Violates:

Prototype: void Icu_GetPulseWidth(Icu_ChannelType Channel);

Table 3-118. Icu_GetPulseWidth Arguments

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

3.10.4 Structs Reference

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

3.10.4.1 Structure Icu_ChannelConfigType

Structure that contains ICU channel configuration.

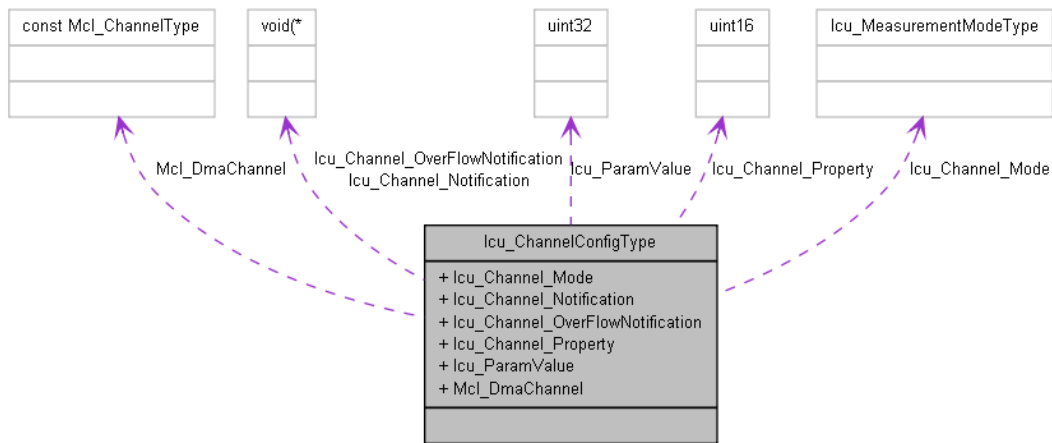


Figure 3-1. Struct Icu_ChannelConfigType

Details:

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

Implements: Icu_ChannelConfigType_struct

Declaration:

```

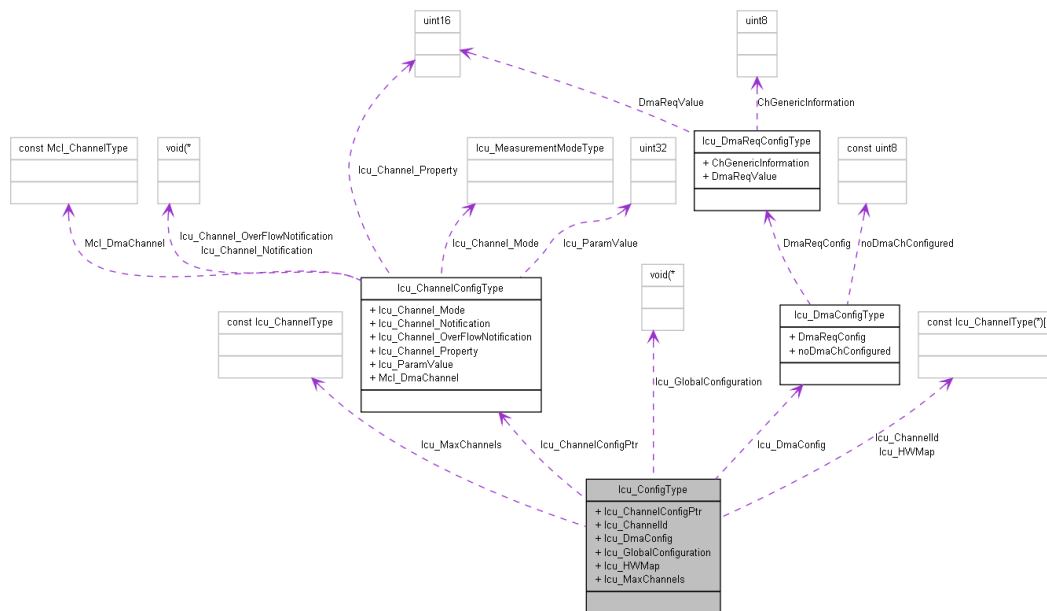
typedef struct
{
    Icu_MeasurementModeType Icu_Channel_Mode,
    Icu_NotifyType Icu_Channel_Notification,
    Icu_NotifyType Icu_Channel_OverFlowNotification,
    Icu_MeasurementSubModeType Icu_Channel_Property,
    Icu_ParamType Icu_ParamValue,
    Icu_WakeupValueType Icu_Channel_WakeupValue,
   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

Table 3-119. Structure Icu_ChannelConfigType member description

Member	Description
Icu_Channel_Mode	The measurement mode for a channel (Timestamp, Signal Measurement, Signal Edge Detect, Edge Counter)
Icu_Channel_Notification	The configured notification for IcuChannel
Icu_Channel_OverflowNotification	The overflow notification for a specific channel
Icu_Channel_Property	The measurement property for a specific channel
Icu_ParamValue	Specific configuration for a channel such as edge polarity, prescalers
Mcl_DmaChannel	Reference to the MCL channel
Icu_Channel_WakeupValue	Reference to the EcuM wakeup source

3.10.4.2 Structure Icu_ConfigType

This type contains initialization data.

**Figure 3-2. Struct Icu_ConfigType**

Details:

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

- Wakeup Module Info (in case the wakeup-capability is true)

- ICU dependent properties for used HW units
- Clock source with optional prescaler (if provided by HW)

Implements: Icu_ConfigType_struct

Declaration:

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

Table 3-120. Structure Icu_ConfigType member description

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

3.10.4.3 Structure Icu_IpConfigType

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

Implements: Icu_IpConfigType_struct

Declaration:

```
typedef struct
{
    Icu_eMios_IpConfigType peMiosIpConfig,
    Icu_Siul2_IpConfigType pSiul2IpConfig,
    Icu_Wkpu_IpConfigType pWkpuIpConfig,
    Icu_IpChannelConfigType (*pIpChannelsConfig) []
} Icu_IpConfigType;
```

Table 3-121. Structure Icu_IpConfigType member description

Member	Description
Icu_eMios_IpConfigType	Pointer to the structure containing eMios configuration
pSiul2IpConfig	Pointer to the structure containing Siul2 configuration
pWkpuIpConfig	Pointer to the structure containing WKPU configuration
pIpChannelsConfig	Pointer to Array containing IP type and index in the IP configuration table for each Icu channel

3.10.4.4 Structure Icu_IpChannelConfigType

Structure that Icu channel high level configuration structure.

Implements: Icu_IpChannelConfigType_struct

Declaration:

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

Table 3-122. Structure Icu_IpChannelConfigType member description

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

3.10.4.5 Structure Icu_eMios_IpConfigType

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

Implements: Icu_eMios_IpConfigType_struct

Declaration:

```
typedef struct
{
    Icu_eMios_ChannelType nNumChannels,
    Icu_eMios_ChannelConfigType (*pChannelsConfig) []
} Icu_eMios_IpConfigType;
```

Table 3-123. Structure Icu_eMios_IpConfigType member description

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

3.10.4.6 Structure Icu_eMios_ChannelConfigType

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

Implements: Icu_eMios_ChannelConfigType_struct

Declaration:

```
typedef struct
{
    Icu_eMios_ChannelType nHwChannel,
    Icu_eMios_ControlType nControlValue,
    uint8 nDefaultStartEdge,
    boolean bSignalMeasure_WithoutInterrupt
} Icu_eMios_ChannelConfigType;
```

Table 3-124. Structure Icu_eMios_ChannelConfigType member description

Member	Description
nHwChannel	Assigned eMios channel id.
nControlValue	eMios channel parameters.
nDefaultStartEdge	eMios Default Start Edge.
bSignalMeasure_WithoutInterrupt	Measurement of ICU signal property without using interrupt.

3.10.4.7 Structure Icu_Siul2_IpConfigType

Structure that combine number of Siul2 channels in the Icu configuration.

Implements: Icu_Siul2_IpConfigType_struct

Declaration:

```
typedef struct
{
    Icu_Siul2_ChannelType nNumChannels,
    Icu_Siul2_GlobalConfigurationType Icu_GlobalConfiguration,
    Icu_Siul2_ChannelConfigType (*pChannelsConfig) []
} Icu_Siul2_IpConfigType;
```

Table 3-125. Structure Icu_Siul2_IpConfigType member description

Member	Description
nNumChannels	Number of Siul2 channels in the Icu configuration
Icu_GlobalConfiguration	The Siul2 global configuration paramters
pChannelsConfig	Pointer to the configured channels for Siul2

3.10.4.8 Structure Icu_Siul2_ChannelConfigType

Structure that combine configuration of each Siul2 channels in the Icu.

Implements: Icu_Siul2_ChannelConfigType_struct

Declaration:

```
typedef struct
{
    Icu_Siul2_ChannelType nHwChannel,
    Icu_Siul2_ControlType nControlValue,
    uint8 nDefaultStartEdge
} Icu_Siul2_ChannelConfigType;
```

Table 3-126. Structure Icu_Siul2_ChannelConfigType member description

Member	Description
nHwChannel	Assigned Siul2 channel id.
nControlValue	Siul2 channel parameters.
nDefaultStartEdge	Siul2 Default Start Edge.

3.10.4.9 Structure Icu_Wkpu_IpConfigType

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

Implements: Icu_Wkpu_IpConfigType_struct

Declaration:

```
typedef struct
{
    Icu_Wkpu_ChannelType nNumChannels,
    Icu_Wkpu_ChannelConfigType (*pChannelsConfig) []
} Icu_Wkpu_IpConfigType;
```

Table 3-127. Structure Icu_Wkpu_IpConfigType member description

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

3.10.4.10 Structure Icu_Wkpu_ChannelConfigType

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

Implements: Icu_Wkpu_ChannelConfigType_struct

Declaration:

```
typedef struct
{
    Icu_Wkpu_ChannelType nHwChannel,
    Icu_Wkpu_ControlType nControlValue,
    uint8 nDefaultStartEdge
} Icu_Wkpu_ChannelConfigType;
```

Table 3-128. Structure Icu_Wkpu_ChannelConfigType member description

Member	Description
nHwChannel	Assigned Wkpu channel id.
nControlValue	Wkpu channel parameters.
nDefaultStartEdge	Wkpu Default Start Edge.

3.10.4.11 Structure Icu_DutyCycleType

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

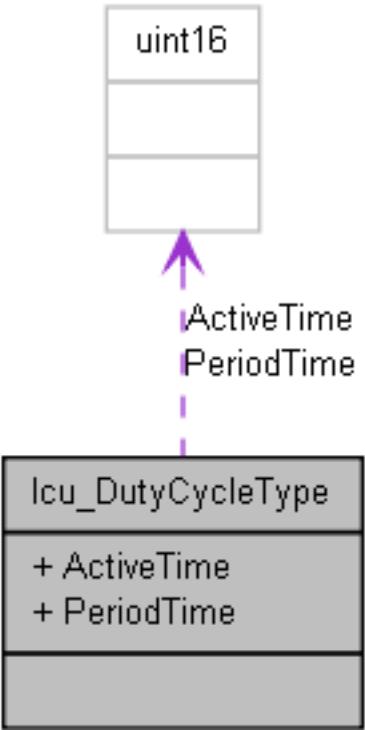


Figure 3-3. Struct Icu_DutyCycleType

Implements: Icu_DutyCycleType_struct

Declaration:

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

Table 3-129. Structure Icu_DutyCycleType member description

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

3.10.5 Types Reference

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

3.10.5.1 Typedef Icu_ChannelType

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

Implements: Icu_ChannelType_typedef

Type: uint16

3.10.5.2 Typedef Icu_EdgeNumberType

Type for saving hardware specific edge number.

Implements: Icu_EdgeNumberType_typedef

Type: Icu_HwSpecificEdgeNumberType

3.10.5.3 Typedef Icu_IndexType

Type for saving the ICU Hardware specific index.

Implements: Icu_IndexType_typedef

Type: Icu_HwSpecificIndexType

3.10.5.4 Typedef Icu_NotifyType

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

Implements: Icu_NotifyType_typedef

Type: void(*)

3.10.5.5 Typedef Icu_ParamType

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

Implements: Icu_ParamType_typedef

Type: uint32

3.10.5.6 Typedef Icu_ValueType

Type for saving the timer register width value.

Implements: Icu_ValueType_typedef

Type: Icu_TimerRegisterWidthType

3.10.5.7 Typedef Icu_WakeupValueType

Type for saving the Wakeup value.

Implements: Icu_WakeupValueType_typedef

Type: uint32

3.10.5.8 Typedef Icu_MeasurementSubModeType

Type for saving the ICU measurement submode type.

Implements: Icu_MeasurementSubModeType_typedef

Type: uint16

3.10.5.9 Typedef Icu_GlobalConfigType

Type for saving the ICU global configuration type.

Implements: Icu_GlobalConfigType_typedef

Type: uint32

3.10.6 Variables

This is a list of all used variables for all modules:

Maximum number of channels

= ICU_MAX_CH=160

Manual

MANUAL =User Manual

Driver Short Name

DRV_NAME = ICU

Driver Module ID

DRV_MODULE_ID = 122

IP Name

IPV_NAME1= eMIOS

IP Name

IPV_NAME2 = SIUL2

IP Name

IPV_NAME3 = WKPU

Driver Full Name

DRV_FULL_NAME = Input Capture Unit

Driver Prefix (used for filenames)

DRV_PREFIX = Icu

NoOfEMIOSs

NO_OF_EMIOSS = 3

NoOfIrqChannels

NO_IRQ = 11

AUTOSAR Specification Document Name

AUTOSAR_SWS_NAME = AUTOSAR_SWS_IcuDriver.pdf

Resource1

RESOURCE_1 = MPC5748G_MAPBGA324

Resource2

RESOURCE_2 = MPC5748G_MAPBGA256

Resource3

RESOURCE_3 = MPC5748G_LQFP176

Resource4

RESOURCE_4 = MPC5748C_MAPBGA324

Resource5

RESOURCE_5 = MPC5748C_MAPBGA256

Resource6

RESOURCE_6 = MPC5748C_LQFP176

Resource7

RESOURCE_7 = MPC5747G_MAPBGA324

Resource8

RESOURCE_8 = MPC5747G_MAPBGA256

Resource9

RESOURCE_9 = MPC5747G_LQFP176

Resource10

RESOURCE_10 = MPC5747C_MAPBGA324

Resource11

RESOURCE_11 = MPC5747C_MAPBGA256

Resource12

RESOURCE_12 = MPC5747C_LQFP176

Resource13

RESOURCE_13 = MPC5746G_MAPBGA324

Resource14

RESOURCE_14 = MPC5746G_MAPBGA256

Resource15

RESOURCE_15 = MPC5746G_LQFP176

Resource16

RESOURCE_16 = MPC5746D_MAPBGA256

Resource17

RESOURCE_17 = MPC5746D_MAPBGA100

Resource18

RESOURCE_18 = MPC5746D_LQFP176

Resource19

RESOURCE_19 = MPC5746C_MAPBGA324

Resource20

RESOURCE_20 = MPC5746C_MAPBGA256

Resource21

RESOURCE_21 = MPC5746C_MAPBGA100

Resource22

RESOURCE_22 = MPC5746C_LQFP176

Resource23

RESOURCE_23 = MPC5746B_MAPBGA256

Resource24

RESOURCE_24 = MPC5746B_MAPBGA100

Resource25

RESOURCE_25 = MPC5746B_LQFP176

Resource26

RESOURCE_26 = MPC5745D_MAPBGA256

Resource27

RESOURCE_27 = MPC5745D_MAPBGA100

Resource28

RESOURCE_28 = MPC5745D_LQFP176

Resource29

RESOURCE_29 = MPC5745C_MAPBGA256

Resource30

RESOURCE_30 = MPC5745C_MAPBGA100

Resource31

RESOURCE_31 = MPC5745C_LQFP176

Resource32

RESOURCE_32 = MPC5745B_MAPBGA256

Resource33

RESOURCE_33 = MPC5745B_MAPBGA100

Resource34

RESOURCE_34 = MPC5745B_LQFP176

This is a list of variables specific for the ICU module:

Input Capture Unit ICU Driver SWS Document Version

SWS_VERSION = 4.2.0 Rev 3

3.11 Symbolic Names Disclaimer

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

```
#define <Container_Short_Name> <Container_ID>
```

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

Chapter 4

Tresos Configuration Plug-in

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

4.1 Configuration elements of Icu

Included forms :

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

Table 4-1. Revision table

Revision	Date
4.2.2	2016-11-25

4.2 Form IMPLEMENTATION_CONFIG_VARIANT

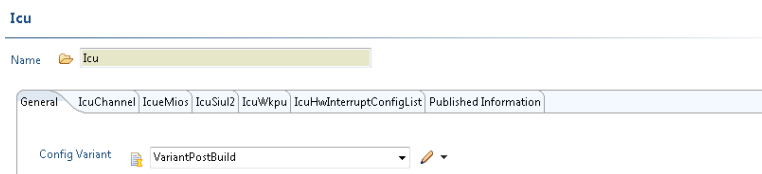


Figure 4-1. Tresos Plugin snapshot for IMPLEMENTATION_CONFIG_VARIANT form.

Table 4-2. Attribute IMPLEMENTATION_CONFIG_VARIANT detailed description

Property	Value
Label	Config Variant
Default	VariantPostBuild
Range	VariantPostBuild VariantPreCompile

4.3 Form IcuGeneral

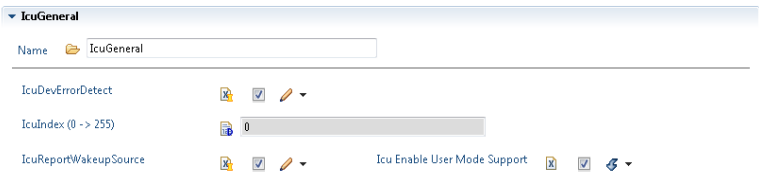


Figure 4-2. Tresos Plugin snapshot for IcuGeneral form.

4.3.1 IcuDevErrorDetect (IcuGeneral)

Switches the Development Error Detection and Notification on or off.

- true: Enabled.
- false: Disabled.

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

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

4.3.2 IcuIndex (IcuGeneral)

Specifies the InstanceId of this module instance.

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

Note

In current implementation this parameter is not used.

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

Property	Value
Type	INTEGER
Origin	AUTOSAR_ECUC
SCOPE	LOCAL
Symbolic Name	false
Default	0
Enable	false
Invalid	Range ≥ 0 ≤ 255

4.3.3 IcuReportWakeupSource (IcuGeneral)

Switch for enabling Wakeup source reporting.

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

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

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

4.3.4 IcuEnableUserModeSupport (IcuGeneral)

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

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

Table 4-6. Attribute IcuEnableUserModeSupport (IcuGeneral) detailed description

Property	Value
Type	BOOLEAN
Symbolic Name	false
Default	false

4.4 Form IcuNonAUTOSAR

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

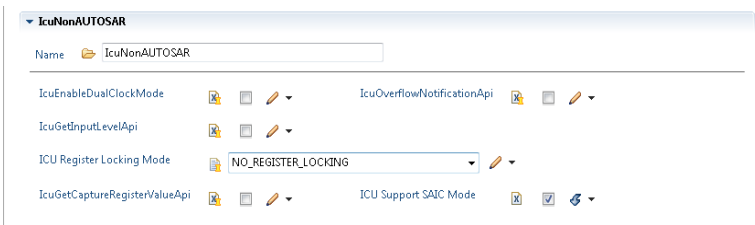


Figure 4-3. TRESOS Plugin snapshot for IcuNonAUTOSAR form.

4.4.1 IcuOverflowNotificationApi (IcuNonAUTOSAR)

Add / removes Overflow Notification functionality.

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

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

Note

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

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

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.4.2 IcuGetInputLevelApi (IcuNonAUTOSAR)

Add / removes Icu_GetInputLevel API from the code.

This function returns Input pin state.

Note

This feature is not required by Autosar.

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

Property	Value
Origin	Custom
Symbolic Name	false
Default	false

4.4.3 IcuRegisterLockingMode (IcuNonAUTOSAR)

IcuRegisterLockingMode can take following values:

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

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

Property	Value
Label	ICU Register Locking Mode
Origin	Custom

Table continues on the next page...

Table 4-9. Attribute IcuRegisterLockingMode (IcuNonAUTOSAR) detailed description (continued)

Property	Value
Symbolic Name	false
Default	NO_REGISTER_LOCKING
Range	NO_REGISTER_LOCKING

4.4.4 IcuEnableDualClockMode (IcuNonAUTOSAR)

Add / removes Icu_SetClockMode API from the code.

This function allows configuration of the dual clock mode.

Note

This feature is not required by Autosar.

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

Property	Value
Origin	Custom
Symbolic Name	false
Default	false

4.4.5 IcuSupportSAICModeApi (IcuNonAUTOSAR)

Enable/Disable SAIC mode for eMios IP.

Note

This feature is not required by Autosar.

Table 4-11. Attribute IcuSupportSAICModeApi (IcuNonAUTOSAR) detailed description

Property	Value
Origin	Custom
Symbolic Name	false
Default	false

4.5 Form IcuOptionalApis

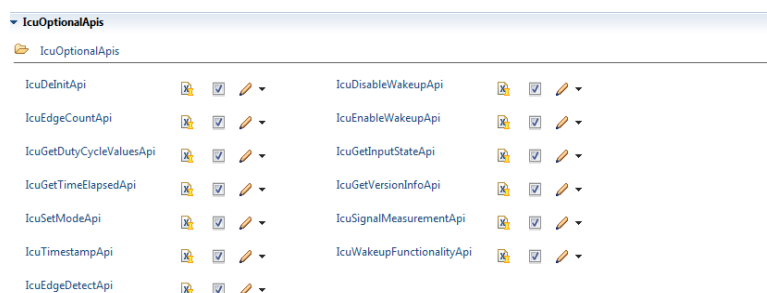


Figure 4-4. TresoS Plugin snapshot for IcuOptionalApis form.

4.5.1 IcuDeInitApi (IcuOptionalApis)

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

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

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

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

4.5.2 IcuDisableWakeupApi (IcuOptionalApis)

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

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

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

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
SCOPE	LOCAL

Table continues on the next page...

Table 4-13. Attribute IcuDisableWakeupApi (IcuOptionalApis) detailed description (continued)

Property	Value
Symbolic Name	false
Default	true

4.5.3 IcuEdgeCountApi (IcuOptionalApis)

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

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

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

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

4.5.4 IcuEnableWakeupApi (IcuOptionalApis)

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

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

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

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

4.5.5 IcuGetDutyCycleValuesApi (IcuOptionalApis)

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

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

Note

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

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

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

4.5.6 IcuGetInputStateApi (IcuOptionalApis)

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

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

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

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

4.5.7 IcuGetTimeElapsedApi (IcuOptionalApis)

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

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

Note

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

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

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

4.5.8 IcuGetVersionInfoApi (IcuOptionalApis)

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

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

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

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

4.5.9 IcuSetModeApi (IcuOptionalApis)

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

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

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

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

4.5.10 IcuSignalMeasurementApi (IcuOptionalApis)

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

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

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

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

4.5.11 IcuTimestampApi (IcuOptionalApis)

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

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

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

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
SCOPE	LOCAL

Table continues on the next page...

Table 4-22. Attribute IcuTimestampApi (IcuOptionalApis) detailed description (continued)

Property	Value
Symbolic Name	false
Default	true

4.5.12 IcuWakeupFunctionalityApi (IcuOptionalApis)

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

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

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

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

4.5.13 IcuEdgeDetectApi (IcuOptionalApis)

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

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

Table 4-24. Attribute IcuEdgeDetectApi (IcuOptionalApis) detailed description

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

4.6 Form CommonPublishedInformation

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

The screenshot shows the 'Resource' configuration window. The 'Published Information' tab is selected. Under the 'CommonPublishedInformation' section, the following fields are visible:

- AUTOSAR Major Version: 4
- AUTOSAR Minor Version: 2
- AUTOSAR Release Revision Version: 2
- Module Id: 0
- Software Major Version: 1
- Software Minor Version: 0
- Software Patch Version: 0
- Vendor Api Infix: (empty)
- Vendor Id: 43

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

4.6.1 ArReleaseMajorVersion (CommonPublishedInformation)

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

Table 4-25. Attribute ArReleaseMajorVersion (CommonPublishedInformation) detailed description

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

4.6.2 ArReleaseMinorVersion (CommonPublishedInformation)

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

Table 4-26. Attribute ArReleaseMinorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	AUTOSAR Minor Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	2
Invalid	Range ≥ 2 ≤ 2

4.6.3 ArReleaseRevisionVersion (CommonPublishedInformation)

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

Table 4-27. Attribute ArReleaseRevisionVersion (CommonPublishedInformation) detailed description

Property	Value
Label	AUTOSAR Release Revision Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	2
Invalid	Range ≥ 2 ≤ 2

4.6.4 ModuleId (CommonPublishedInformation)

Module ID of this module from Module List.

Table 4-28. Attribute ModuleId (CommonPublishedInformation) detailed description

Property	Value
Label	Module Id
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	122
Invalid	Range >=122 <=122

4.6.5 SwMajorVersion (CommonPublishedInformation)

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

Table 4-29. Attribute SwMajorVersion (CommonPublishedInformation) detailed description

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

4.6.6 SwMinorVersion (CommonPublishedInformation)

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

Table 4-30. Attribute SwMinorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	Software Minor Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false

Table continues on the next page...

Table 4-30. Attribute SwMinorVersion (CommonPublishedInformation) detailed description (continued)

Property	Value
Default	0
Invalid	Range >=0 <=0

4.6.7 SwPatchVersion (CommonPublishedInformation)

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

Table 4-31. Attribute SwPatchVersion (CommonPublishedInformation) detailed description

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

4.6.8 VendorApiInfix (CommonPublishedInformation)

In driver modules which can be instantiated several times on a single ECU, BSW00347 requires that the name of APIs is extended by the VendorId and a vendor specific name. This parameter is used to specify the vendor specific name. In total, the implementation specific name is generated as follows:

<ModuleName>_>VendorId>_<VendorApiInfix><Api name from SWS>. E.g. assuming that the VendorId of the implementor is 123 and the implementer chose a VendorApiInfix of "v11r456" a api name Can_Write defined in the SWS will translate to Can_123_v11r456Write. This parameter is mandatory for all modules with upper multiplicity > 1. It shall not be used for modules with upper multiplicity =1.

Table 4-32. Attribute VendorApiInfix (CommonPublishedInformation) detailed description

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

4.6.9 VendorId (CommonPublishedInformation)

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

Table 4-33. Attribute VendorId (CommonPublishedInformation) detailed description

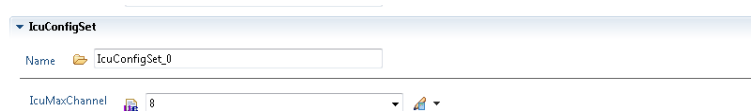
Property	Value
Label	Vendor Id
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	43
Invalid	Range >=43 <=43

4.7 Form IcuConfigSet

This container is the base for a multiple configuration set

Included forms :

- [Form IcuChannel](#)

**Figure 4-6. Tresos Plugin snapshot for IcuConfigSet form.**

4.7.1 IcuMaxChannel (IcuConfigSet)

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

Note

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

Table 4-34. Attribute IcuMaxChannel (IcuConfigSet) detailed description

Property	Value
Type	INTEGER
Origin	AUTOSAR_ECUC
SCOPE	LOCAL
Symbolic Name	false
Invalid	Range >0 <=160

4.7.2 Form IcuChannel

Is included by form : [Form IcuConfigSet](#)

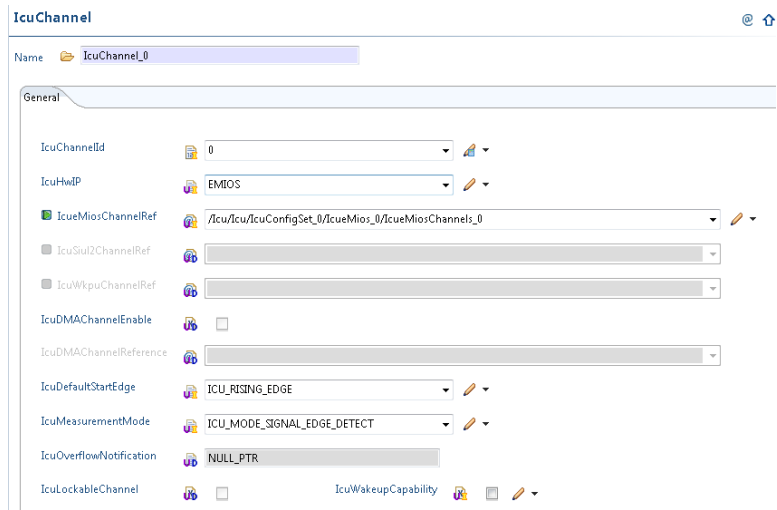


Figure 4-7. Tresos Plugin snapshot for IcuChannel form.

4.7.2.1 IcuChannelId (IcuChannel)

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

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

Property	Value
Type	INTEGER
Origin	AUTOSAR_ECUC
Symbolic Name	true
Invalid	Range >=0 <43

4.7.2.2 IcuHwIP (IcuChannel)

Selects the physical HwChannelId Channel.

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

Property	Value
Type	ENUMERATION
Origin	Custom
DEFAULT	EMIOS
RANGE	EMIOS SIUL2 WKPU

4.7.2.3 IcueMiosChannelRef (IcuChannel)

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

Reference to the EMIOS Channel.

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

Property	Value
Type	REFERENCE
Origin	Custom

4.7.2.4 IcuSiul2ChannelRef (IcuChannel)

Select the SIUL2 channel on which the functionality of the current ICU channel will be implemented

Reference to the SIUL2 Channel.

Table 4-38. Attribute IcuSiul2ChannelRef (IcuChannel) detailed description

Property	Value
Type	REFERENCE
Origin	Custom

4.7.2.5 IcuWkupChannelRef (IcuChannel)

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

Reference to the WKPU Channel.

Table 4-39. Attribute IcuWkupChannelRef (IcuChannel) detailed description

Property	Value
Type	REFERENCE
Origin	Custom

4.7.2.6 IcuDMAChannelEnable (IcuChannel)

IcuLockableChannel

indicates if the corresponding channel will use DMA for measurement

Table 4-40. Attribute IcuDMAChannelEnable (IcuChannel) detailed description

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.7.2.7 IcuDefaultStartEdge (IcuChannel)

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

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

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

Implementation Type: Icu_ActivationType

Table 4-41. Attribute IcuDefaultStartEdge (IcuChannel) detailed description

Property	Value
Type	ENUMERATION
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	ICU_RISING_EDGE
Range	ICU_BOTH_EDGES ICU_FALLING_EDGE ICU_RISING_EDGE

4.7.2.8 IcuMeasurementMode (IcuChannel)

Configures the measurement mode of this channel.

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

Implementation Type: Icu_MeasurementModeType

Table 4-42. Attribute IcuMeasurementMode (IcuChannel) detailed description

Property	Value
Type	ENUMERATION
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	ICU_MODE_SIGNAL_EDGE_DETECT
Range	ICU_MODE_EDGE_COUNTER ICU_MODE_SIGNAL_EDGE_DETECT ICU_MODE_SIGNAL_MEASUREMENT ICU_MODE_TIMESTAMP

4.7.2.9 IcuOverflowNotification (IcuChannel)

Icu Overflow Notification Handler

In order to activate this field you have to:

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

Note

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

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

Table 4-43. Attribute IcuOverflowNotification (IcuChannel) detailed description

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

4.7.2.10 IcuLockableChannel (IcuChannel)

IcuLockableChannel

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

Table 4-44. Attribute IcuLockableChannel (IcuChannel) detailed description

Property	Value
Type	BOOLEAN

Table continues on the next page...

Table 4-44. Attribute IcuLockableChannel (IcuChannel) detailed description (continued)

Property	Value
Origin	Custom
Symbolic Name	false
Default	false

4.7.2.11 IcuWakeupCapability (IcuChannel)

Information about the wakeup-capability of this channel.

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

Table 4-45. Attribute IcuWakeupCapability (IcuChannel) detailed description

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	false

4.7.2.12 IcuDMAChannelRef (IcuChannel)

Icu DMA Channel Reference

Reference to the DMA Channel configure for the Request

Table 4-46. Attribute IcuDMAChannelRef (IcuChannel) detailed description

Property	Value
Label	IcuDMAChannelReference
Type	CHOICE-REFERENCE
Origin	Custom

4.7.2.13 Form IcuSignalEdgeDetection

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

Is included by form : [Form IcuChannel](#)

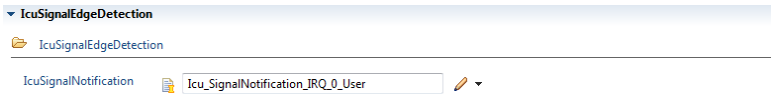


Figure 4-8. Tresos Plugin snapshot for IcuSignalEdgeDetection form.

4.7.2.13.1 IcuSignalNotification (IcuSignalEdgeDetection)

Notification function for signal notification.

Table 4-47. Attribute IcuSignalNotification (IcuSignalEdgeDetection) detailed description

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

4.7.2.14 Form IcuSignalMeasurement

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

Is included by form : [Form IcuChannel](#)

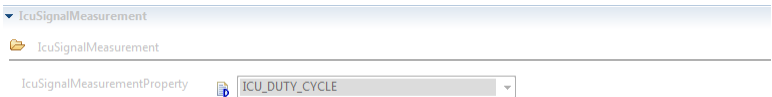


Figure 4-9. Tresos Plugin snapshot for IcuSignalMeasurement form.

4.7.2.14.1 IcuSignalMeasurementProperty (IcuSignalMeasurement)

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

This property can not be changed during runtime.

Followings are measurement mode

- ICU_DUTY_CYCLE

- ICU_HIGH_TIME
- ICU_LOW_TIME
- ICU_PERIOD_TIME

Implementation Type: Icu_SignalMeasurementPropertyType

Table 4-48. Attribute IcuSignalMeasurementProperty (IcuSignalMeasurement) detailed description

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

4.7.2.15 Form IcuTimestampMeasurement

Is included by form : [Form IcuChannel](#)

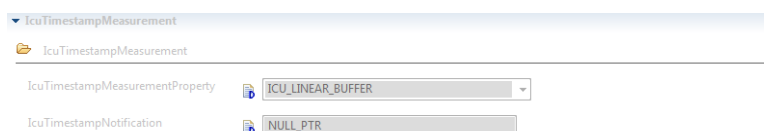


Figure 4-10. Tresos Plugin snapshot for IcuTimestampMeasurement form.

4.7.2.15.1 IcuTimestampMeasurementProperty (IcuTimestampMeasurement)

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

- ICU_CIRCULAR_BUFFER.
- ICU_LINEAR_BUFFER.

Implementation Type: Icu_TimestampBufferType

Table 4-49. Attribute IcuTimestampMeasurementProperty (IcuTimestampMeasurement) detailed description

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

4.7.2.15.2 IcuTimestampNotification (IcuTimestampMeasurement)

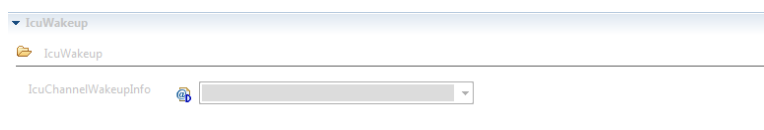
Notification function if the number of requested timestamps(Notification interval > 0) are acquired.

Table 4-50. Attribute IcuTimestampNotification (IcuTimestampMeasurement) detailed description

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

4.7.2.16 Form IcuWakeup

Is included by form : [Form IcuChannel](#)

**Figure 4-11. Tresos Plugin snapshot for IcuWakeup form.**

4.7.2.16.1 IcuChannelWakeupInfo (IcuWakeup)

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

Implementation Type: reference to EcuM_WakeupSourceType

Table 4-51. Attribute IcuChannelWakeupInfo (IcuWakeup) detailed description

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

4.7.3 Form IcueMios

Is included by form : [Form IcuConfigSet](#)

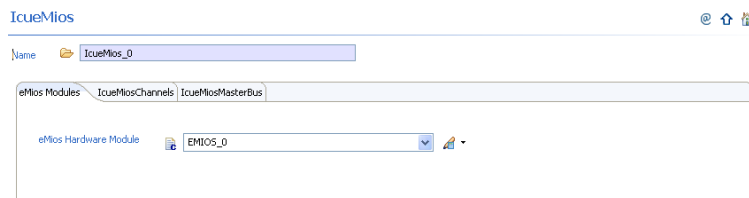


Figure 4-12. Tresos Plugin snapshot for IcueMios form.

4.7.3.1 IcueMiosModule (IcueMios)

Select the physical eMios Module.

Table 4-52. Attribute IcuChannelId (IcueMios) detailed description

Property	Value
Type	ENUMERATION
Origin	Custom
Symbolic Name	false

4.7.3.2 Form IcueMiosChannel

Is included by form : [Form IcueMios](#)

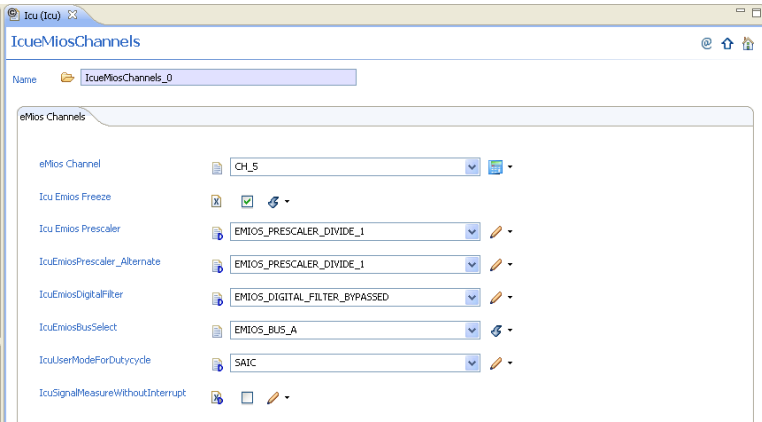


Figure 4-13. Tressos Plugin snapshot for IcueMiosChannel form.

4.7.3.2.1 IcueMiosChannel (IcueMiosChannel)

Select the physical eMios Module.

Table 4-53. Attribute IcueMiosChannel (IcueMiosChannel) detailed description

Property	Value
Type	ENUMERATION
Origin	Custom
Symbolic Name	false

4.7.3.2.2 IcuEmiosFreeze (IcueMiosChannel)

If selected eMIOS channel registers are freezed in debug mode.

Table 4-54. Attribute IcuEmiosFreeze (IcueMiosChannel) detailed description

Property	Value
Label	Icu Emios Freeze
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.7.3.2.3 IcuEmiosPrescaler (IcueMiosChannel)

If an eMIOS channel is being used, this parameter configures the clock divider value for the internal prescaler of specific Unified Channel.

Note

This is an Implementation Specific Parameter.

This parameter will be editable only when use EMIOS_BUS_INTERNAL_COUNTER.

Table 4-55. Attribute IcuEmiosPrescaler (IcueMiosChannel) detailed description

Property	Value
Label	Icu Emios Prescaler
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	EMIOS_PRESCALER_DIVIDE_1
Invalid	Range EMIOS_PRESCALER_DIVIDE_1 EMIOS_PRESCALER_DIVIDE_2 EMIOS_PRESCALER_DIVIDE_3 EMIOS_PRESCALER_DIVIDE_4

4.7.3.2.4 IcuEmiosPrescaler_Alternate(IcueMiosChannel)

IcuEmiosPrescaler_Alternate: If an eMIOS channel is being used, this parameter configures the alternate clock divider value for the internal prescaler of specific Unified Channel.

Note

This is an Implementation Specific Parameter.

This parameter will be editable only when use EMIOS_BUS_INTERNAL_COUNTER.

Table 4-56. Attribute IcuEmiosPrescaler_Alternate (IcueMiosChannel) detailed description

Property	Value
Label	IcuEmiosPrescaler_Alternate
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	EMIOS_PRESCALER_DIVIDE_1
Invalid	Range EMIOS_PRESCALER_DIVIDE_1 EMIOS_PRESCALER_DIVIDE_2 EMIOS_PRESCALER_DIVIDE_3 EMIOS_PRESCALER_DIVIDE_4

4.7.3.2.5 IcuEmiosDigitalFilter(IcucMiosChannel)

IcuEmiosDigitalFilter: If a eMIOS channel is being used this option is active, possible values are: 0 (Bypassed), 2, 4, 8, 16 FLT_Clock periods. This parameter configures programmable input filter. It selects the minimum input pulse width [FLT_CLK periods] that can pass through the filter.

Note

This is an Implementation Specific Parameter.

Table 4-57. Attribute IcuEmiosDigitalFilter (IcucMiosChannel) detailed description

Property	Value
Label	IcuEmiosDigitalFilter
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	EMIOS_DIGITAL_FILTER_BYPASSED
Invalid	Range EMIOS_DIGITAL_FILTER_BYPASSED EMIOS_DIGITAL_FILTER_02 EMIOS_DIGITAL_FILTER_04 EMIOS_DIGITAL_FILTER_08 EMIOS_DIGITAL_FILTER_16

4.7.3.2.6 IcuEmiosBusSelect(IcucMiosChannel)

IcuEmiosBusSelect: Selects the counter used with the unified channel

Note

This is an Implementation Specific Parameter.

Table 4-58. Attribute IcuEmiosBusSelect (IcucMiosChannel) detailed description

Property	Value
Label	IcuEmiosBusSelect
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	EMIOS_BUS_A
Invalid	Range EMIOS_BUS_A EMIOS_BUS_F EMIOS_BUS_DIVERSE EMIOS_BUS_INTERNAL_COUNTER

4.7.3.2.7 IcuUserModeForDutycycle (IcucMiosChannel)

Selection of the signal measurement mode when IcuSignalMeasurementProperty is ICU_DUTY_CYCLE.

This parameter will be enabled in configuration only when IcuSignalMeasurementProperty is ICU_DUTY_CYCLE. The following channels support IPM and IPWM mode. Advantages of IPM,IPWM mode over SAIC

The size of the driver code to measure duty cycle is less for a channel which supports IPWM mode as compared to a channel which uses SAIC mode to capture the duty cycle.

For every period of the measured signal, the interrupt generated is just 1 in IPWM mode as compared to a channel which uses SAIC mode where the number of interrupt generated is 2. Because of these reasons the driver code execution speed is faster in IPWM mode as compared to SAIC mode

If we configure the EMIOS channel which uses IPWM mode to capture a duty cycle of a varying PWM Singal, we will get the interrupt during only one edge i.e opposite edge of the starting edge. But to capture the duty cycle of a varying PWM Signal, we have to get the interrupt during both falling edge and rising edge. Hence in IPWM mode we get the delayed duty cycle values Because of this limitation of IPWM mode, we can use EMIOS channel which uses SAIC mode to capture the duty cycle of a varying PWM Signal.

Table 4-59. Attribute IcuUserModeForDutycycle (IcucMiosChannel) detailed description

Property	Value
Type	IMPLEMENTATIONCONFIGCLASS
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	SAIC
Range	SAIC IPWM

4.7.3.2.8 IcuSignalMeasureWithoutInterrupt (IcucMiosChannel)

Icu Signal measurement without using interrupt.

- true: Without interrupt.
- false: With interrupt.

Table 4-60. Attribute IcuSignalMeasureWithoutInterrupt (IcucMiosChannel) detailed description

Property	Value
Type	BOOLEAN
Origin	Custom
SYMBOLICNAMEVALUE	false
DEFAULT	false

4.7.3.3 Form IcucMiosMasterBus

Is included by form : [Form IcucMios](#)

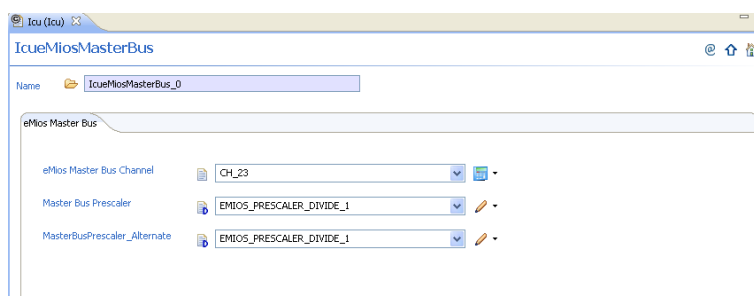


Figure 4-14. Tressos Plugin snapshot for IcucMiosMasterBus form.

4.7.3.3.1 eMiosMasterBus (IcucMiosMasterBus)

Selects one of the eMIOS channels available on the platform.

Table 4-61. Attribute eMiosMasterBus (IcucMiosMasterBus) detailed description

Property	Value
Type	ENUMERATION
Origin	Custom
Symbolic Name	false

4.7.3.3.2 MasterBusPrescaler (IcucMiosMasterBus)

If an eMIOS channel is being used, this parameter configures the clock divider value for the internal prescaler of specific Unified Channel.

Note

This is an Implementation Specific Parameter.

Table 4-62. Attribute MasterBusPrescaler (IcucMiosMasterBus) detailed description

Property	Value
Label	Icu Emios Prescaler
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	EMIOS_PRESCALER_DIVIDE_1
Invalid	Range EMIOS_PRESCALER_DIVIDE_1 EMIOS_PRESCALER_DIVIDE_2 EMIOS_PRESCALER_DIVIDE_3 EMIOS_PRESCALER_DIVIDE_4

4.7.3.3.3 MasterBusPrescaler_Alternate (IcucMiosMasterBus)

MasterBusPrescaler_Alternate: If an eMIOS channel is being used, this parameter configures the alternate clock divider value for the internal prescaler of specific Unified Channel.

Note

This is an Implementation Specific Parameter.

Table 4-63. Attribute MasterBusPrescaler_Alternate (IcucMiosMasterBus) detailed description

Property	Value
Label	IcuEmiosPrescaler_Alternate
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	EMIOS_PRESCALER_DIVIDE_1
Invalid	Range EMIOS_PRESCALER_DIVIDE_1 EMIOS_PRESCALER_DIVIDE_2 EMIOS_PRESCALER_DIVIDE_3 EMIOS_PRESCALER_DIVIDE_4

4.7.4 Form IcuSiul2

Is included by form : [Form IcuConfigSet](#)

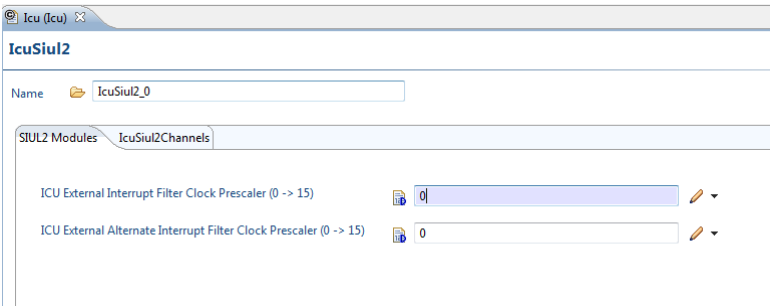


Figure 4-15. Tresos Plugin snapshot for IcuSiul2 form.

4.7.4.1 ICU External Interrupt Filter Clock Prescaler (IcuSiul2)

Configure the clock prescaler which is used to select the clock for all digital filter counters in the SIUL2.

Note

: This is an Implementation Specific Parameter.

Table 4-64. Attribute ICU External Interrupt Filter Clock Prescaler (IcuSiul2) detailed description

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

4.7.4.2 ICU External Alternate Interrupt Filter Clock Prescaler (IcuSiul2)

Configure the clock alternate prescaler which is used to select the clock for all digital filter counters in the SIUL2. This is only available when Dual Clock mode is activated.

Note

: This is an Implementation Specific Parameter.

Table 4-65. Attribute ICU External Alternate Interrupt Filter Clock Prescaler (IcuSiul2) detailed description

Property	Value
Type	INTEGER
Origin	Custom
Symbolic Name	false
DEFAULT	0
Invalid	Range >0 <=15

4.7.4.3 Form IcuSiul2Channel

Is included by form : [Form IcuSiul2](#)

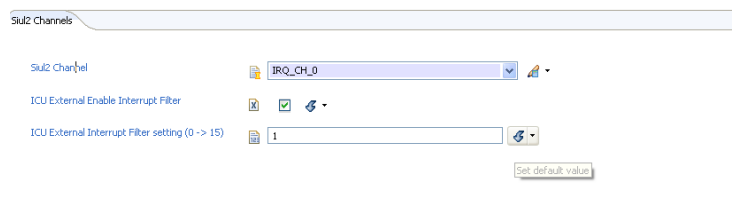


Figure 4-16. Tresos Plugin snapshot for IcuSiul2Channel form.

4.7.4.3.1 IcuSiul2Channel (IcuSiul2Channel)

Selects one of the Siul2 channels available on the platform.

Table 4-66. Attribute IcuSiul2Channel (IcuSiul2Channel) detailed description

Property	Value
Type	ENUMERATION
Origin	Custom
Symbolic Name	false

4.7.4.3.2 Icu_EXT_ISR_IFERDigitalFilter (IcuSiul2Channel)

Enable external digital filter counter on the interrupt pads to filter out glitches on the inputs

Note

This is an Implementation Specific Parameter.

Table 4-67. Attribute Icu_EXT_ISR_IFERDigitalFilter (IcuSiul2Channel) detailed description

Property	Value
Label	ICU External Enable Interrupt Filter
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.7.4.3.3 Icu_EXT_ISR_IFMCDigitalFilter (IcuSiul2Channel)

Maximum Interrupt Filter Counter setting.

Note

This is an Implementation Specific Parameter.

Table 4-68. Attribute Icu_EXT_ISR_IFMCDigitalFilter (IcuSiul2Channel) detailed description

Property	Value
Label	ICU External Interrupt Filter setting
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	0
Invalid	Range >0 <=15

4.7.5 Form IcuWkpu

Is included by form : [Form IcuConfigSet](#)

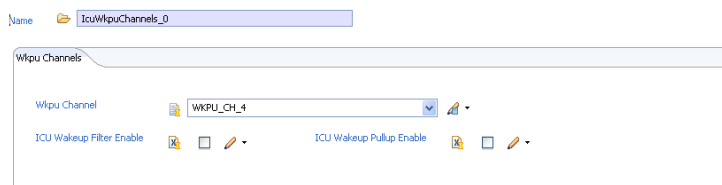


Figure 4-17. Tressos Plugin snapshot for IcuWkpu form.

4.7.5.1 IcuWkpuChannel (IcuWkpu)

Selects one of the Wkpu channels available on the platform.

Table 4-69. Attribute IcuWkpuChannel (IcuWkpu) detailed description

Property	Value
Type	ENUMERATION
Origin	Custom
Symbolic Name	false

4.7.5.2 IcuWKPU_ISR_WIPUER (IcuWkpu)

Maximum Interrupt Filter Counter setting.

Note

This is an Implementation Specific Parameter.

Table 4-70. Attribute IcuWKPU_ISR_WIPUER (IcuWkpu) detailed description

Property	Value
Label	ICU Wakeup Pullup Enable
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.7.5.3 Icu_EXT_ISR_WIFERDigitalFilter (IcuWkpu)

Enable external digital filter counter on the interrupt pads to filter out glitches on the inputs.

Note

This is an Implementation Specific Parameter.

Table 4-71. Attribute Icu_EXT_ISR_WIFERDigitalFilter (IcuWkpu) detailed description

Property	Value
Label	ICU Wakeup Filter Enable
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.7.6 Form IcuHwInterruptConfigList

Is included by form : [Form IcuConfigSet](#)

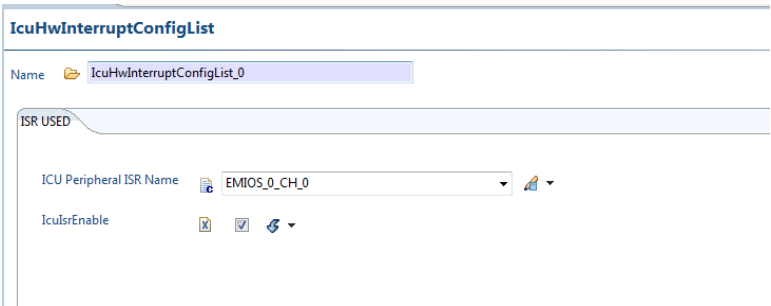


Figure 4-18. Tresos Plugin snapshot for IcuHwInterruptConfigList form.

4.7.6.1 ICU Peripheral ISR Name (IcuHwInterruptConfigList)

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

Note

: This is an Implementation Specific Parameter.

Table 4-72. ICU Peripheral ISR Name(IcuHwInterruptConfigList) detailed description

Property	Value
Type	ENUMERATION
Origin	Custom
Symbolic Name	true

4.7.6.2 IculsrEnable(IcuHwInterruptConfigList)

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

Note

This is an Implementation Specific Parameter.

If user forget tick on corresponding channel checkbox which he configured then it is the root cause of crash interrupt handling in eMios IP.

Table 4-73. Attribute IculsrEnable (IcuHwInterruptConfigList) detailed description

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
DEFAULT	false

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 μ 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.

© 2012–2017 NXP B.V.

Document Number UM351CUASR4.2
Rev0002RTM1.0.0
Revision 5.0.0