
User Manual

for MPC5634M ADC Driver

Document Number: UM14ADCASR3.0R2.0.0

Rev. 1.2





Contents

Section Number	Title	Page
Chapter 1		
Revision History		
Chapter 2		
Introduction		
2.1	Supported Derivatives.....	11
2.2	Overview.....	11
2.3	About this Manual.....	12
2.4	Acronyms and Definitions.....	12
2.5	Reference List.....	13
Chapter 3		
ADC Driver		
3.1	Requirements.....	15
3.2	Driver Design Summary.....	15
3.3	Deviation from Requirements.....	16
3.4	Runtime Errors.....	17
3.5	Software specification.....	18
3.5.1	Define Reference.....	18
3.5.2	Enum Reference.....	18
3.5.3	Function Reference.....	18
3.5.3.1	Function Adc_DeInit.....	18
3.5.3.2	Function Adc_DisableGroupNotification.....	19
3.5.3.3	Function Adc_DisableHardwareTrigger.....	19
3.5.3.4	Function Adc_EnableGroupNotification.....	20
3.5.3.5	Function Adc_EnableHardwareTrigger.....	21
3.5.3.6	Function Adc_GetGroupStatus.....	22
3.5.3.7	Function Adc_GetStreamLastPointer.....	22
3.5.3.8	Function Adc_GetVersionInfo.....	23
3.5.3.9	Function Adc_Init.....	24

Section Number	Title	Page
3.5.3.10	Function Adc_ReadGroup.....	25
3.5.3.11	Function Adc_SetMode.....	26
3.5.3.12	Function Adc_SetupResultBuffer.....	26
3.5.3.13	Function Adc_StartGroupConversion.....	27
3.5.3.14	Function Adc_StopGroupConversion.....	28
3.5.3.15	Function Adc_Dma_Interrupt_Common_Func.....	29
3.5.3.16	Function Adc_FIFO_Disable_Func.....	30
3.5.3.17	Function Adc_Irq_CFIFO_Empty_Common_Func.....	32
3.5.3.18	Function Adc_Irq_RFIFO_Full_Common_Func.....	33
3.5.3.19	Function Adc_DisableChannel.....	34
3.5.3.20	Function Adc_DisableHwTrigger.....	35
3.5.3.21	Function Adc_EnableChannel.....	35
3.5.3.22	Function Adc_EnableHwTrigger.....	36
3.5.3.23	Function Adc_HwResultReadGroup.....	37
3.5.3.24	Function ADC_SetClockMode.....	38
3.5.4	Structs Reference.....	38
3.5.4.1	Structure Adc_ConfigType.....	39
3.5.4.2	Structure Adc_GroupStatusType.....	39
3.5.4.3	Structure Adc_LLD_ChannelConfiguration.....	40
3.5.4.4	Structure Adc_LLD_GroupConfiguration.....	41
3.5.4.5	Structure Adc_LLD_HWUnitConfiguration.....	42
3.5.4.6	Structure Adc_LLD_MultiConfig.....	43
3.5.4.7	Structure Adc_UnitStatusType.....	45
3.5.5	Types Reference.....	45
3.5.5.1	Typedef Adc_ChannelType.....	46
3.5.5.2	Typedef Adc_ClockSourceType.....	46
3.5.5.3	Typedef Adc_ConversionTimeType.....	46
3.5.5.4	Typedef Adc_GroupDefType.....	46
3.5.5.5	Typedef Adc_GroupPriorityType.....	47

Section Number	Title	Page
3.5.5.6	Typedef Adc_GroupType.....	47
3.5.5.7	Typedef Adc_HwTriggerTimerType.....	47
3.5.5.8	Typedef Adc_HwUnitType.....	48
3.5.5.9	Typedef Adc_NotifyType.....	48
3.5.5.10	Typedef Adc_PrescaleType.....	48
3.5.5.11	Typedef Adc_QueueIndexType.....	49
3.5.5.12	Typedef Adc_ResolutionType.....	49
3.5.5.13	Typedef Adc_SamplingTimeType.....	49
3.5.5.14	Typedef Adc_StreamNumSampleType.....	49
3.5.5.15	Typedef Adc_ValueGroupType.....	50
3.5.5.16	Typedef Adc_VoltageSourceType.....	50
3.5.5.17	Typedef Adc_WdgNotifyType.....	50
3.5.5.18	Typedef EQADC_fifo_Dma_Channel.....	51
3.5.6	Variables Reference.....	51
3.5.6.1	Variable Adc_Cfg_Ptr.....	51
3.5.6.2	Variable ADC_GroupStatus.....	51
3.5.6.3	Variable ADC_UnitStatus.....	51
3.6	Symbolic Names Disclaimer.....	52

Chapter 4 Tresos Configuration Plug-in

4.1	Configuration elements of Adc.....	53
4.2	Form IMPLEMENTATION_CONFIG_VARIANT.....	53
4.3	Form AdcGeneral.....	54
4.3.1	AdcDeInitApi (AdcGeneral).....	54
4.3.2	AdcDevErrorDetect (AdcGeneral).....	55
4.3.3	AdcEnableStartStopGroupApi (AdcGeneral).....	55
4.3.4	AdcGrpNotifCapability (AdcGeneral).....	56
4.3.5	AdcHwTriggerApi (AdcGeneral).....	56
4.3.6	AdcVersionInfoApi (AdcGeneral).....	56

Section Number	Title	Page
4.3.7	AdcReadGroupApi (AdcGeneral).....	57
4.3.8	AdcPriorityImplementation (AdcGeneral).....	57
4.3.9	AdcEnableQueuing (AdcGeneral).....	58
4.3.10	AdcCalibrationApi (AdcGeneral).....	58
4.3.11	AdcTimeout (AdcGeneral).....	59
4.4	Form AdcPublishedInformation.....	59
4.4.1	AdcChannelValueSigned (AdcPublishedInformation).....	60
4.4.2	AdcGroupFirstChannelFixed (AdcPublishedInformation).....	60
4.4.3	AdcMaxChannelResolution (AdcPublishedInformation).....	60
4.5	Form CommonPublishedInformation.....	61
4.5.1	ArMajorVersion (CommonPublishedInformation).....	61
4.5.2	ArMinorVersion (CommonPublishedInformation).....	62
4.5.3	ArPatchVersion (CommonPublishedInformation).....	62
4.5.4	ModuleId (CommonPublishedInformation).....	63
4.5.5	SwMajorVersion (CommonPublishedInformation).....	63
4.5.6	SwMinorVersion (CommonPublishedInformation).....	64
4.5.7	SwPatchVersion (CommonPublishedInformation).....	64
4.5.8	VendorId (CommonPublishedInformation).....	65
4.5.9	VendorApiInfix (VendorApiInfix).....	65
4.6	Form NonAutosar.....	66
4.6.1	AdcChIndexSymNames (NonAutosar).....	66
4.6.2	AdcReadGroupOptimization (NonAutosar).....	67
4.7	Form AdcConfigSet.....	67
4.7.1	AdcDigitalFilterLength (AdcConfigSet).....	68
4.7.2	Form AdcGeneric.....	68
4.7.2.1	AdcPriorityQueueMaxDepth (AdcGeneric).....	69
4.7.2.2	AdcMaxGroupChannels (AdcGeneric).....	69
4.7.2.3	AdcTransferType (AdcGeneric).....	70

Section Number	Title	Page
4.7.3	Form AdcHwUnit.....	70
4.7.3.1	AdcClockSource (AdcHwUnit).....	71
4.7.3.2	AdcHwUnitId (AdcHwUnit).....	71
4.7.3.3	AdcPrescale (AdcHwUnit).....	72
4.7.3.4	AdcCalibration (AdcHwUnit).....	72
4.7.3.5	AdcExternalMultiplexing (AdcHwUnit).....	73
4.7.3.6	Form AdcChannel.....	73
4.7.3.6.1	AdcChannelConvTime (AdcChannel).....	74
4.7.3.6.2	AdcChannelId (AdcChannel).....	74
4.7.3.6.3	AdcHwChannel (AdcChannel).....	75
4.7.3.6.4	AdcChannelRefVoltsrcHigh (AdcChannel).....	75
4.7.3.6.5	AdcChannelRefVoltsrcLow (AdcChannel).....	76
4.7.3.6.6	AdcChannelResolution (AdcChannel).....	76
4.7.3.6.7	AdcChannelCalibration (AdcChannel).....	77
4.7.3.6.8	AdcChannelSampTime (AdcChannel).....	77
4.7.3.7	Form AdcGroup.....	78
4.7.3.7.1	AdcGroupAccessMode (AdcGroup).....	78
4.7.3.7.2	AdcGroupConversionMode (AdcGroup).....	79
4.7.3.7.3	AdcGroupId (AdcGroup).....	79
4.7.3.7.4	AdcGroupPriority (AdcGroup).....	80
4.7.3.7.5	AdcGroupReplacement (AdcGroup).....	80
4.7.3.7.6	AdcGroupTriggSrc (AdcGroup).....	81
4.7.3.7.7	AdcHwTrigSignal (AdcGroup).....	81
4.7.3.7.8	AdcHwTrigTimer (AdcGroup).....	82
4.7.3.7.9	AdcNotification (AdcGroup).....	82
4.7.3.7.10	AdcStreamingBufferMode (AdcGroup).....	83
4.7.3.7.11	AdcResultBufferPointer (AdcGroup).....	83
4.7.3.7.12	AdcStreamingNumSamples (AdcGroup).....	83
4.7.3.7.13	AdcHwTrigSrc (AdcGroup).....	84

Section Number	Title	Page
4.7.3.7.14	AdcGroupFifo (AdcGroup).....	84
4.7.3.7.15	AdcGroupISRAlignSamples (AdcGroup).....	85
4.7.3.7.16	AdcGroupDefinition (AdcGroupDefinition).....	85

Chapter 1

Revision History

Table 1-1. Revision History

Revision	Date	Author	Description
1.0	03-Feb-2011	Alfredo Di Martino	Update for Monaco automatic documentation
1.1	02-Aug-2011	Alfredo Di Martino	Update for Monaco HF automatic documentation
1.2	19-Dec-2011	Alfredo Di Martino	Updated for Monaco RTM 2.0.0



Chapter 2

Introduction

This User Manual describes Freescale Semiconductor AUTOSAR Driver Full Name (ShortName) for MPC5634M .

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

2.1 Supported Derivatives

The software described in this document is intended to be used with the following microcontroller devices of Freescale Semiconductor .

Table 2-1. MPC5634M Derivatives

Freescale Semiconductor	mpc5634m_bga208, mpc5634m_qfp144, mpc5634m_qfp176
-------------------------	--

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

2.2 Overview

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

AUTOSAR

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

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

2.3 About this Manual

This Technical Reference employs the following typographical conventions:

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

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

Notes and warnings are shown as below:

Note

This is a note.

2.4 Acronyms and Definitions

Table 2-2. Acronyms and Definitions

Term	Definition
ADC	Analog to Digital Converter
API	Application Programming Interface
ASM	Assembler
AUTOSAR	Automotive Open System Architecture
BSMI	Basic Software Make file Interface
CAN	Controller Area Network
C/CPP	C and C++ Source Code
CS	Chip Select
CTU	Cross Trigger Unit
DEM	Diagnostic Event Manager
DET	Development Error Tracer

Table continues on the next page...

Table 2-2. Acronyms and Definitions (continued)

Term	Definition
DMA	Direct Memory Access
ECU	Electronic Control Unit
FIFO	First In First Out
LSB	Least Significant Bit
MCU	Micro Controller Unit
MIDE	Multi Integrated Development Environment
MSB	Most Significant Bit
N/A	Not Applicable
RAM	Random Access Memory
SIU	Systems Integration Unit
SWS	Software Specification
VLE	Variable Length Encoding
XML	Extensible Markup Language

2.5 Reference List

Table 2-3. Reference List

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

Chapter 3

ADC Driver

3.1 Requirements

Requirements for this driver are detailed in the AUTOSAR 3.0 ShortName Driver Software Specification document (See Table [Reference List](#)).

3.2 Driver Design Summary

The ADC Driver initializes and controls the internal Analogue to Digital Converter Unit(s) of the microcontroller. It provides services to start and stop a conversion respectively to enable and disable the trigger source for a conversion. Furthermore it provides services to enable and disable a notification mechanism and routines to query the status and result of a conversion. The ADC Driver shall work on so called ADC Channels. An ADC channel combines an analogue input pin, the needed ADC circuitry itself and a conversion result register into an entity that can be individually controlled and accessed via the ADC Driver. The driver provides a service for Streaming management results and for De-Initialization of circuits.

Moreover the following ADC units hardware channels will be used:

- **MPC5634M device families at 144 pins:**
 - Adc Physical Channels for ADC HW Unit 0 and ADC HW Unit 1:
AN_0:AN_35;AN_38,AN_39,VRH_40,VRL_41,VRH_MINUS_VRL_BY_2_42,
VRH_MINUS_VRL_75PE_43,VRH_MINUS_VRL_25PE_44,INA_ADC01_0_45;
ANW_64:ANW_71;ANX_72:ANX_79;ANY_80:ANY_87;ANZ_88:ANZ_95;
DAN0PL_DAN0MI_96,DAN1PL_DAN1MI_97,DAN2PL_DAN2MI_98,DAN3PL_DAN3MI_99,

INA_ADC01_1_128,INA_ADC01_2_129,INA_ADC0_3_162,INA_ADC0_4_163,
 INA_ADC0_5_164,INA_ADC0_6_165,INA_ADC0_7_166,INA_ADC0_8_167,
 INA_ADC1_3_194,INA_ADC1_4_195,INA_ADC1_5_196,INA_ADC1_6_197,
 INA_ADC1_7_198,INA_ADC1_8_199;ANR_224:ANR_231;ANS_232:ANS_239;
 ANT_240:ANT_247.

- **MPC5634M device families at 176 pins:**

- Adc Physical Channels for ADC HW Unit 0 and ADC HW Unit 1:
 AN_0:AN_39,VRH_40,VRL_41,VRH_MINUS_VRL_BY_2_42,
 VRH_MINUS_VRL_75PE_43,VRH_MINUS_VRL_25PE_44,INA_ADC01_0_45;
 ANW_64:ANW_71;ANX_72:ANX_79;ANY_80:ANY_87;ANZ_88:ANZ_95;DAN0PL_DAN0MI_96,
 DAN1PL_DAN1MI_97,DAN2PL_DAN2MI_98,DAN3PL_DAN3MI_99,INA_ADC01_1_128,
 INA_ADC01_2_129,INA_ADC0_3_162,INA_ADC0_4_163,INA_ADC0_5_164,
 INA_ADC0_6_165,INA_ADC0_7_166,INA_ADC0_8_167,INA_ADC1_3_194,
 INA_ADC1_4_195,INA_ADC1_5_196,INA_ADC1_6_197,INA_ADC1_7_198,
 INA_ADC1_8_199;ANR_224:ANR_231;ANS_232:ANS_239;ANT_240:ANT_247.

3.3 Deviation from Requirements

The driver deviates from the AUTOSAR ShortName Driver software specification in some places. The table identifies the AUTOSAR requirements that are not fully implemented, implemented differently, or out of scope for the ShortName Driver. Table [Table 3-1](#) provides Status column description.

Table 3-1. Deviations Status Column Description

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

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

Table 3-2. ADC Deviations Table

Requirement	Status	Description	Notes
ADC345	N/I	The ADC module priority mechanism shall allow suspending and resuming of channel group conversion.	For this version has been implemented the abort/restart mechanism.
ADC341, ADC339, ADC337	N/A	Priority mechanism supported by hardware.	Hw queue are not supported by hardware.
ADC319	N/F	In streaming access mode the result buffer shall have $m \times n$ elements, where n is the numbers of channels belonging to the group, m the number of the sample acquired per channel.	The implementation is as per $n \times m$ as shown in the Figure 2 of the 3.0ShortName (version V2.2.0 R3.0 Rev 0001) [Reference List]: Example for Group and Result Buffer configuration of Autosar SWS but autosar ADC319 requirement is inconsistent with this figure.

3.4 Runtime Errors

The driver generates the following DEM errors at runtime.

Table 3-3. Runtime Errors

Function	Error Code	Condition triggering the error
Adc_Init()	ADC_E_TIMEOUT;	Timeout expired. ADC hardware is not entered in power down state.
Adc_Init()	ADC_E_TIMEOUT;	Timeout expired. ADC hardware is not entered in idle state.
Adc_StartGroupConversion()	ADC_E_TIMEOUT;	Timeout expired. Ongoing conversion is not aborted.
Adc_StopGroupConversion()	ADC_E_TIMEOUT;	Timeout expired. Ongoing conversion is not aborted.
Adc_EnableHardwareTrigger()	ADC_E_TIMEOUT;	Timeout expired. ADC hardware is not entered in power down state.
Adc_EnableHwTrigger()	ADC_E_TIMEOUT;	Timeout expired. ADC hardware is not entered in power down state.
Adc_SetMode()	ADC_E_TIMEOUT;	Timeout expired. ADC hardware is not entered in power down state.
Adc_SetMode()	ADC_E_TIMEOUT;	Timeout expired. ADC hardware is not entered in idle state.
Adc_FIFO_Disable_Func()	ADC_E_TIMEOUT;	Timeout expired. ADC hardware fails to disable the FIFO's of eQADC_A.
Adc_EnableHardwareTrigger()	ADC_E_TIMEOUT;	Timeout expired. double buffered registers are not updated.
Adc_DisableHardwareTrigger()	ADC_E_TIMEOUT;	Timeout expired. double buffered registers are not updated.

3.5 Software specification

The following sections contains driver software specifications.

3.5.1 Define Reference

Constants supported by the driver are as per AUTOSAR ShortName Driver software specification Version 3.0 .

3.5.2 Enum Reference

Enumeration of all constants supported by the driver are as per AUTOSAR ShortName Driver software specification Version 3.0 .

3.5.3 Function Reference

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

3.5.3.1 Function Adc_Delnit

Returns all ADC HW Units to a state comparable to their power on reset state.

Details:

Returns all ADC HW Units to a state comparable to their power on reset state, and de-initialize the ADC MCAL driver.

Note

The function Autosar Service ID[hex]: 0x01. Synchronous. Non Reentrant function.

Satisfied Requirements: ADC110, ADC111, ADC112, ADC154, ADC228, ADC366.

Note

Violates MISRA 2004 Advisory Rule 8.10, external function could be made static

- See `Adc_c_REF_3`

Prototype: `void Adc_DeInit(void);`

3.5.3.2 Function `Adc_DisableGroupNotification`

Disables the notification mechanism for the requested ADC Channel group.

Details:

This function will disable the notification mechanism only for the requested ADC channel group.

Note

The function Autosar Service ID[hex]: 0x08. Synchronous.
Reentrant function.

Satisfied Requirements: ADC058, ADC101, ADC131, ADC166, ADC300, ADC373.

Note

Violates MISRA 2004 Advisory Rule 8.10, external function could be made static

- See `Adc_c_REF_3`

Prototype: `void Adc_DisableGroupNotification(Adc_GroupType Group);`

Table 3-4. `Adc_DisableGroupNotification` Arguments

Type	Name	Direction	Description
<code>Adc_GroupType</code>	Group	input	Numeric ID of requested ADC channel group.

3.5.3.3 Function `Adc_DisableHardwareTrigger`

Disables the hardware trigger for the requested ADC Channel group.



Figure 3-1. Inverse references of fuction Adc_DisableHardwareTrigger

Details:

This function will disable the HW trigger source for the requested ADC channel group.

Note

The function Autosar Service ID[hex]: 0x06. Synchronous.
Reentrant function.

Satisfied Requirements: ADC116, ADC121, ADC129, ADC137, ADC145, ADC157, ADC266, ADC282, ADC298, ADC304, ADC361, ADC371, ADC429.

Note

Violates MISRA 2004 Advisory Rule 8.10, external function
could be made static

- See `Adc_c_REF_3`

Prototype: `void Adc_DisableHardwareTrigger(Adc_GroupType Group);`

Table 3-5. Adc_DisableHardwareTrigger Arguments

Type	Name	Direction	Description
Adc_GroupType	Group	input	Numeric ID of requested ADC channel group.

3.5.3.4 Function Adc_EnableGroupNotification

Enables the notification mechanism for the requested ADC Channel group.

Details:

This function will enable the notification mechanism only for the requested ADC channel group.

Note

The function Autosar Service ID[hex]: 0x07. Synchronous.
Reentrant function.

Satisfied Requirements: ADC057, ADC100, ADC130, ADC165, ADC299, ADC372.

Note

Violates MISRA 2004 Advisory Rule 8.10, external function could be made static

- See `Adc_c_REF_3`

Prototype: `void Adc_EnableGroupNotification(Adc_GroupType Group);`

Table 3-6. Adc_EnableGroupNotification Arguments

Type	Name	Direction	Description
Adc_GroupType	Group	input	Numeric ID of requested ADC channel group.

3.5.3.5 Function Adc_EnableHardwareTrigger

Enables the hardware trigger for the requested ADC Channel group.



Figure 3-2. Function Adc_EnableHardwareTrigger References.

Details:

This function will enable the HW trigger source for the requested ADC channel group. This function does set the CTU register for all platform that have the CTU Hw Unit.

Note

The function Autosar Service ID[hex]: 0x05. Synchronous.
Reentrant function.

Satisfied Requirements: ADC114, ADC120, ADC128, ADC136, ADC144, ADC281, ADC297, ADC321, ADC349, ADC265, ADC353, ADC370, ADC425, ADC432, ADC273.

Note

Violates MISRA 2004 Advisory Rule 8.10, external function could be made static

- See `Adc_c_REF_3`

Prototype: `void Adc_EnableHardwareTrigger(Adc_GroupType Group);`

Table 3-7. Adc_EnableHardwareTrigger Arguments

Type	Name	Direction	Description
Adc_GroupType	Group	input	Numeric ID of requested ADC channel group.

3.5.3.6 Function Adc_GetGroupStatus

Returns the conversion status of the requested ADC Channel group.

Details:

This function will return the converison status of the requested ADC Channel group.

Note

The function Autosar Service ID[hex]: 0x09. Synchronous.
Reentrant function.

Return: Adc_StatusType Conversion status for the requested group.

Satisfied Requirements: ADC220, ADC221, ADC222, ADC224, ADC225, ADC226, ADC301, ADC305, ADC325, ADC374.

Note

Violates MISRA 2004 Advisory Rule 8.10, external function
could be made static

- See Adc_c_REF_3

Prototype: Adc_StatusType Adc_GetGroupStatus(Adc_GroupType Group);

Table 3-8. Adc_GetGroupStatus Arguments

Type	Name	Direction	Description
Adc_GroupType	Group	input	Numeric ID of requested ADC channel group.

Table 3-9. Adc_GetGroupStatus Return Values

Name	Description
ADC_IDLE	In case of errors.
Conversion	Status else.

3.5.3.7 Function `Adc_GetStreamLastPointer`

Returns the number of valid samples per channel.

Details:

Returns the number of valid samples per channel, stored in the result buffer. Reads a pointer, pointing to a position in the group result buffer. With the pointer position, the results of all group channels of the last completed conversion round can be accessed. With the pointer and the return value, all valid group conversion results can be accessed (the user has to take the layout of the result buffer into account).

Note

The function Autosar Service ID[hex]: 0x0b. Synchronous.
Reentrant function.

Return: `Adc_StreamNumSampleType` Number of valid samples per channel.

Satisfied Requirements: ADC214, ADC215, ADC216, ADC218, ADC302, ADC326, ADC327, ADC328, ADC375, ADC382, ADC387, ADC219, ADC418.

Note

Violates MISRA 2004 Advisory Rule 8.10, external function
could be made static

- See `Adc_c_REF_3`

Prototype: `Adc_StreamNumSampleType Adc_GetStreamLastPointer(Adc_GroupType Group, Adc_ValueGroupType **PtrToSamplePtr);`

Table 3-10. `Adc_GetStreamLastPointer` Arguments

Type	Name	Direction	Description
<code>Adc_GroupType</code>	Group	input	Numeric ID of requested ADC channel group.
<code>Adc_ValueGroupType **</code>	PtrToSamplePtr	output	Pointer to result buffer pointer.

Table 3-11. `Adc_GetStreamLastPointer` Return Values

Name	Description
0	In case of errors.
Number	Of valid samples per channel.

3.5.3.8 Function Adc_GetVersionInfo

Returns the version information of this module.

Note

The function Autosar Service ID[hex]: 0x0A. Synchronous.
Non Reentrant function.

Satisfied Requirements: ADC236, ADC237, ADC324, ADC376.

Note

Violates MISRA 2004 Advisory Rule 8.10, external function
could be made static

- See `Adc_c_REF_3`

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

Table 3-12. Adc_GetVersionInfo Arguments

Type	Name	Direction	Description
Std_VersionInfoType *	versioninfo	output	Pointer to where to store the version information of this module.

3.5.3.9 Function Adc_Init

Initializes the ADC hardware unit and the driver.

Details:

This function will initialize both the ADC HW unit and the driver structures.

Note

The function Autosar Service ID[hex]: 0x00. Synchronous. Non
Reentrant function.

Satisfied Requirements: ADC054, ADC056, ADC077, ADC107, ADC246, ADC247,
ADC248, ADC249, ADC250, ADC307, ADC342, ADC343, ADC344, ADC365.

Note

Violates MISRA 2004 Advisory Rule 8.10, external function
could be made static

- See `Adc_c_REF_3`

Prototype: `void Adc_Init(const Adc_ConfigType *ConfigPtr);`

Table 3-13. Adc_Init Arguments

Type	Name	Direction	Description
const Adc_ConfigType *	ConfigPtr	input	Pointer to configuration set in Variant PB (Variant PC requires a NULL_PTR).

3.5.3.10 Function Adc_ReadGroup

Reads the group conversion results.

Details:

Reads the group conversion results of the last completed conversion round of the requested group and stores the channel values starting at the DataBufferPtr address. The group channel values are stored in ascending channel number order (in contrast to the storage layout of the result buffer if streaming access is configured).

Note

The function Autosar Service ID[hex]: 0x04. Synchronous.
Reentrant function.

Return: Std_ReturnType Standard return type.

Satisfied Requirements: ADC075, ADC113, ADC122, ADC152, ADC296, ADC329, ADC330, ADC331, ADC359, ADC369, ADC388.

Note

Violates MISRA 2004 Advisory Rule 8.10, external function
could be made static

- See Adc_c_REF_3

Prototype: `Std_ReturnType Adc_ReadGroup(Adc_GroupType Group, Adc_ValueGroupType *DataPtr);`

Table 3-14. Adc_ReadGroup Arguments

Type	Name	Direction	Description
Adc_GroupType	Group	input	Numeric ID of requested ADC Channel group.
Adc_ValueGroupType *	DataPtr	input	ADC result buffer pointer is initialized with the value of the data buffer pointer.

Table 3-15. Adc_ReadGroup Return Values

Name	Description
E_OK:	Results are available and written to the data buffer.
E_NOT_OK:	No results are available or development error occurred.

3.5.3.11 Function Adc_SetMode

Set the ADC mode either to powerdown or normal.

Note

The function Non Autosar Service ID[hex]: 0x10. Non Reentrant function.

Return: - Std_ReturnType - E_OK or E_NOT_OK.

Satisfied Requirements: :

Note

Violates MISRA 2004 Advisory Rule 8.10, external function could be made static

- See `Adc_c_REF_3`

Prototype: `Std_ReturnType Adc_SetMode(Adc_SetModeType SetMode);`

Table 3-16. Adc_SetMode Arguments

Type	Name	Direction	Description
Adc_SetModeType	SetMode	input	ADC_NORMAL_MODE, ADC_POWER_DOWN_MODE .

Table 3-17. Adc_SetMode Return Values

Name	Description
-	E_OK - in case of successfull settings.
-	E_NOT_OK - in case of unsuccessfull settings.

3.5.3.12 Function Adc_SetupResultBuffer

Initializes the group specific ADC result buffer pointer as configured to point to the DataBufferPtr address which is passed as parameter.

Details:

Initializes the group specific ADC result buffer pointer as configured to point to the DataBufferPtr address which is passed as parameter. The ADC driver stores all group conversion results to result buffer addressed with the result buffer pointer.

Adc_SetupResultBuffer determines the address of the result buffer. After reset, before a group conversion can be started, an initialization of the ADC result buffer pointer is required.

Pre: ADC422.

Note

The function Autosar Service ID[hex]: 0x0c. Synchronous.
Reentrant function.

Return: Std_ReturnType Standard return type.

Satisfied Requirements: ADC419, ADC420, ADC421, ADC422, ADC423.

Note

Violates MISRA 2004 Advisory Rule 8.10, external function
could be made static

- See Adc_c_REF_3

Prototype: Std_ReturnType Adc_SetupResultBuffer(Adc_GroupType Group, Adc_ValueGroupType *DataBufferPtr);

Table 3-18. Adc_SetupResultBuffer Arguments

Type	Name	Direction	Description
Adc_GroupType	Group	input	Numeric ID of requested ADC channel group.
Adc_ValueGroupType *	DataBufferPtr	input	ADC result buffer pointer is initialized with the value of the data buffer pointer.

Table 3-19. Adc_SetupResultBuffer Return Values

Name	Description
E_OK:	Result buffer pointer initialized correctly.
E_NOT_OK:	Operation failed or development error occurred.

3.5.3.13 Function Adc_StartGroupConversion

Starts the conversion of all channels of the requested ADC Channel group.

Details:

This function will start the SW conversion of all channels of the requested ADC channel group.

Note

The function Autosar Service ID[hex]: 0x02. Asynchronous.
Reentrant function.

Satisfied Requirements: ADC061, ADC125, ADC133, ADC146, ADC156, ADC259, ADC294, ADC346, ADC348, ADC351, ADC367, ADC424, ADC426, ADC427, ADC428, ADC431.

Note

Violates MISRA 2004 Advisory Rule 8.10, external function could be made static

- See Adc_c_REF_3

Prototype: void Adc_StartGroupConversion(Adc_GroupType Group);

Table 3-20. Adc_StartGroupConversion Arguments

Type	Name	Direction	Description
Adc_GroupType	Group	input	Numeric ID of requested ADC channel group.

3.5.3.14 Function Adc_StopGroupConversion

Stops the conversion of all channels of the requested ADC Channel group.

Details:

This function will stop the SW conversion of all channels of the requested ADC channel group.

Note

The function Autosar Service ID[hex]: 0x03. Synchronous.
Reentrant function.

Satisfied Requirements: ADC126, ADC155, ADC164, ADC241, ADC260, ADC283, ADC295, ADC360, ADC368, ADC385, ADC386.

Note

Violates MISRA 2004 Advisory Rule 8.10, external function could be made static

- See `Adc_c_REF_3`

Prototype: `void Adc_StopGroupConversion(Adc_GroupType Group);`

Table 3-21. Adc_StopGroupConversion Arguments

Type	Name	Direction	Description
Adc_GroupType	Group	input	Numeric ID of requested ADC channel group.

3.5.3.15 Function Adc_Dma_Interrupt_Common_Func

This function implements the ISR for the conversion done from RFIFO on the HW unit0 or unit1 of eQADC_A.

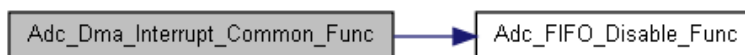


Figure 3-3. Function Adc_Dma_Interrupt_Common_Func References.

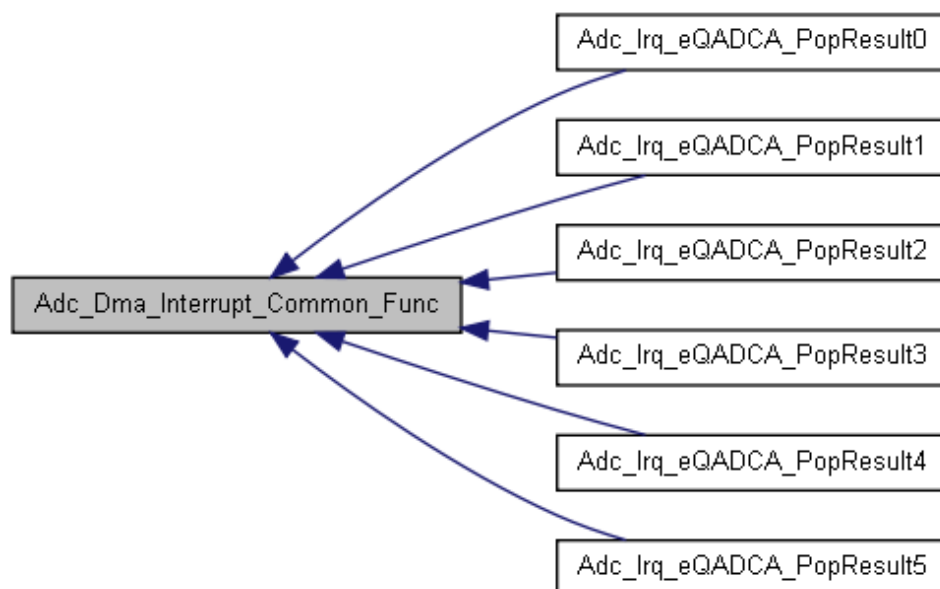


Figure 3-4. Inverse references of function Adc_Dma_Interrupt_Common_Func

Details:

The function implements the ISR for the HW unit0 or unit1 of eQADC_A.

Return: None.

Pre: None.

Post: None.

Prototype: void Adc_Dma_Interrupt_Common_Func(Adc_HwUnitType unit, uint32 eQADC_AorB, uint32 AdcFifoindex);

Table 3-22. Adc_Dma_Interrupt_Common_Func Arguments

Type	Name	Direction	Description
uint32	eQADC_AorB	input	AdcFifoindex.
uint32	eQADC_AorB	input	Always ADC_EQADC_A.
uint32	AdcFifoindex	input	0 to 5.
	None.	output	.
	None.	input, output	.

3.5.3.16 Function Adc_FIFO_Disable_Func

This function is called from the ISR to disable the FIFO's of eQADC_A.

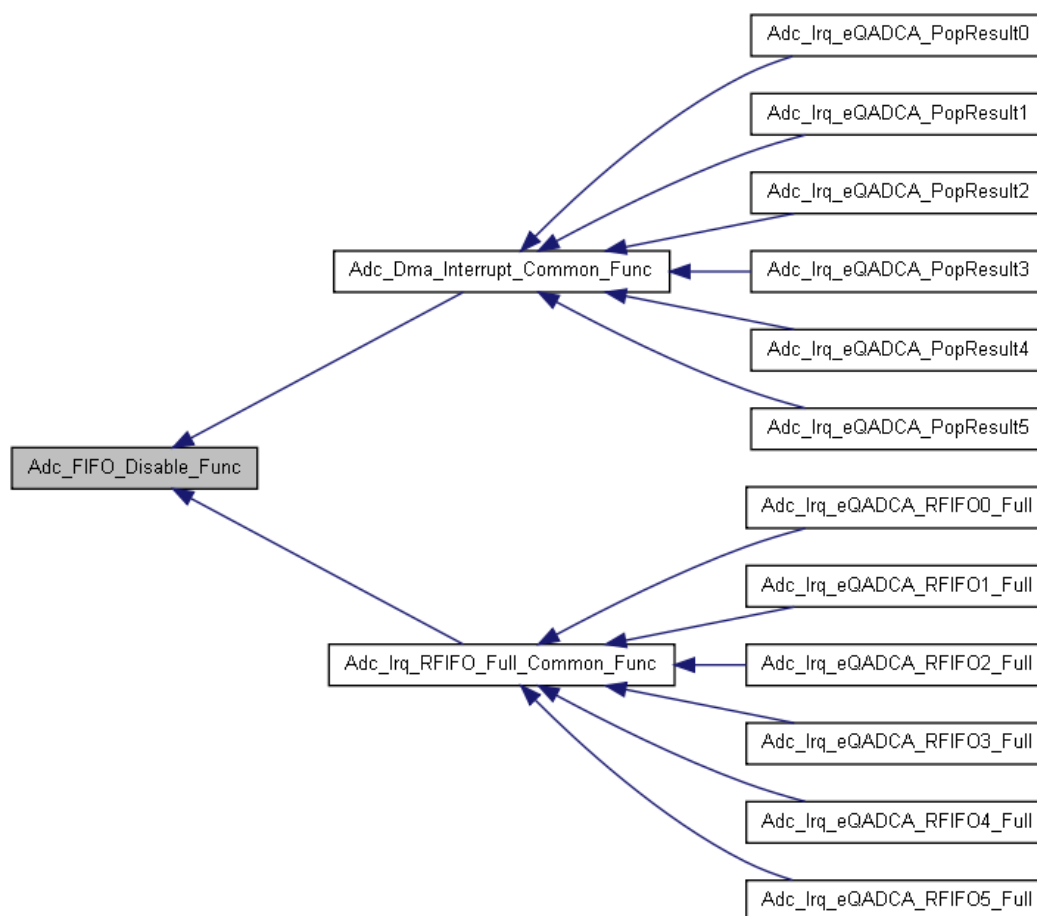


Figure 3-5. Inverse references of fuction Adc_FIFO_Disable_Func

Details:

This function is called from the ISR to disable the FIFO's of eQADC_A.

Return: Std_ReturnType Standard return type.

Pre: None.

Post: None.

Test for FIFO's Status failure.

Report production error and Return from function.

Test for FIFO's status failure.

Report production error and Return from function.

Prototype: Std_ReturnType Adc_FIFO_Disable_Func(uint32 eQADC_AorB, uint32 AdcFifoindex);

Table 3-23. Adc_FIFO_Disable_Func Arguments

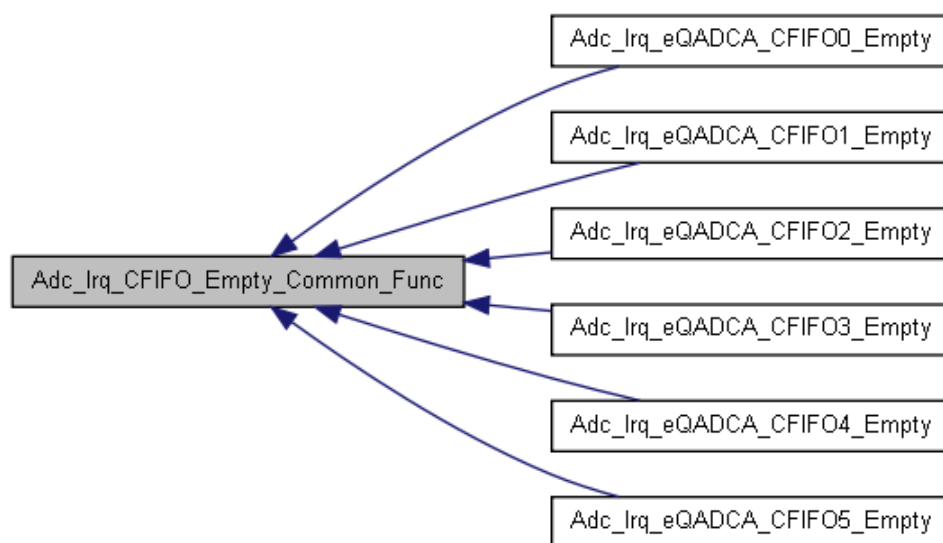
Type	Name	Direction	Description
uint32	eQADC_AorB	input	AdcFifoindex.
uint32	eQADC_AorB	input	Always ADC_EQADC_A.
uint32	AdcFifoindex	input	0 to 5.
	None.	output	.
	None.	input, output	.

Table 3-24. Adc_FIFO_Disable_Func Return Values

Name	Description
E_OK:	Return on successful functionality execution.
E_NOT_OK:	Return when the functionality is failed to execute due to production error.

3.5.3.17 Function Adc_Irq_CFIFO_Empty_Common_Func

This function implements the ISR is called when CFIFO is not full on the HW unit0 or unit1 of eQADC_A.

**Figure 3-6. Inverse references of function Adc_Irq_CFIFO_Empty_Common_Func**

Details:

The function implements the ISR for the HW unit0 or unit1 of eQADC_A.

Return: None.

Pre: None.

Post: None.

Prototype: `void Adc_Irq_CFIFO_Empty_Common_Func(Adc_HwUnitType unit, uint32 eQADC_AorB, uint32 AdcFifoindex);`

Table 3-25. Adc_Irq_CFIFO_Empty_Common_Func Arguments

Type	Name	Direction	Description
uint32	eQADC_AorB	input	AdcFifoindex.
uint32	eQADC_AorB	input	Always ADC_EQADC_A.
uint32	AdcFifoindex	input	0 to 5.
	None.	output	.
	None.	input, output	.

3.5.3.18 Function Adc_Irq_RFIFO_Full_Common_Func

This function implements the ISR is called when RFIFO have data for the HW unit0 or unit1 of eQADC_A.

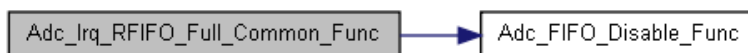


Figure 3-7. Function Adc_Irq_RFIFO_Full_Common_Func References.

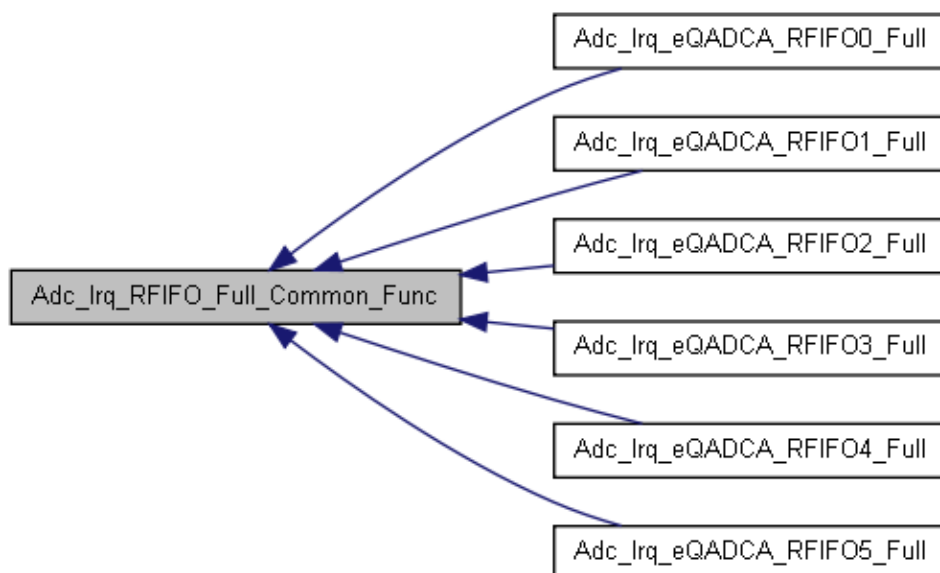


Figure 3-8. Inverse references of fuction Adc_Irq_RFIFO_Full_Common_Func

Details:

The function implements the ISR is called when RFIFO5 have data for the HW unit0 or unit1 of eQADC.

Return: None.

Pre: None.

Post: None.

Prototype: `void Adc_Irq_RFIFO_Full_Common_Func(Adc_HwUnitType unit, uint32 eQADC_AorB, uint32 AdcFifoindex);`

Table 3-26. Adc_Irq_RFIFO_Full_Common_Func Arguments

Type	Name	Direction	Description
uint32	eQADC_AorB	input	AdcFifoindex.
uint32	eQADC_AorB	input	Always ADC_EQADC_A.
uint32	AdcFifoindex	input	0 to 5.
	None.	output	.
	None.	input, output	.

3.5.3.19 Function Adc_DisableChannel

Disable a channel inside a group.

Details:

This function allows to de-active a channel assigned to a group for SW normal conversion

Note

The function Service ID[hex]: 0x13. Synchronous. Reentrant function. Violates Adc_NonASR_c_REF_3: External function could be made static.

Return: void .

Satisfied Requirements: PR-MCAL-3233

Implements:

Prototype: void Adc_DisableChannel(Adc_GroupType GroupId, Adc_ChannelType ChannelId);

Table 3-27. Adc_DisableChannel Arguments

Type	Name	Direction	Description
Adc_GroupType	GroupId	input	Symbolic name of group.
Adc_ChannelType	ChannelId	input	Symbolic name of channel.

3.5.3.20 Function Adc_DisableHwTrigger

Function disable the TriggerSource for group selected by Group parameter.

Details:

Function disable the TriggerSource for group selected by 'Group' parameter.

This non autosar API is used to disable the already enabled TriggerSource of the Group.

Note

The function Service ID[hex]: 0x0F. Violates
Adc_NonASR_c_REF_3: External function could be made static.

This non autosar API is used to disable the already enabled TriggerSource of the Group.

Note

The function Service ID[hex]: 0x0F.

Prototype: void Adc_DisableHwTrigger(Adc_GroupType Group, Adc_HwTriggerTimerType TriggerSource);

Table 3-28. Adc_DisableHwTrigger Arguments

Type	Name	Direction	Description
Adc_GroupType	Group	input	Index of group .
Adc_HwTriggerTimerType	TriggerSource	input	Trigger source to be disable for the group. (Configuration file should contain the TriggerSource for called group).
Adc_GroupType	Group	input	Index of group .
Adc_HwTriggerTimerType	TriggerSource	input	Trigger source to be disable for the group. (Configuration file should contain the TriggerSource for called group).

3.5.3.21 Function `Adc_EnableChannel`

Enable a channel inside a group.

Details:

This function allows to active a channel assigned to a group for SW normal conversion

Note

The function Service ID[hex]: 0x12. Synchronous. Reentrant function. Violates `Adc_NonASR_c_REF_3`: External function could be made static.

Return: void .

Satisfied Requirements: PR-MCAL-3233

Implements:

Prototype: `void Adc_EnableChannel(Adc_GroupType GroupId, Adc_ChannelType ChannelId);`

Table 3-29. `Adc_EnableChannel` Arguments

Type	Name	Direction	Description
<code>Adc_GroupType</code>	<code>GroupId</code>	input	Symbolic name of group.
<code>Adc_ChannelType</code>	<code>ChannelId</code>	input	Symbolic name of channel.

3.5.3.22 Function `Adc_EnableHwTrigger`

Function enables the `TriggerSource` for group selected by `Group` parameter.

Details:

Function enables the `TriggerSource` for group selected by "Group" parameter.

This non autosar API is used to enable any one of the configured `TriggerSource` of the Group. When this non autosar API is used to enable the trigger source the CTU interrupt will be disabled by the driver. So user has to call the non autosar API `Adc_HwResultReadGroup` to read the converted result from the ADC hardware register.

Note

The function Service ID[hex]: 0x0E. Violates `Adc_NonASR_c_REF_3`: External function could be made static.

This non autosar API is used to enable any one of the configured TriggerSource of the Group. When this non autosar API is used to enable the trigger source the CTU interrupt will be disabled by the driver. So user has to call the non autosar API

Adc_HwResultReadGroup to read the converted result from the ADC hardware register.

Note

The function Service ID[hex]: 0x0E.

Prototype: void Adc_EnableHwTrigger(Adc_GroupType Group, Adc_HwTriggerTimerType TriggerSource);

Table 3-30. Adc_EnableHwTrigger Arguments

Type	Name	Direction	Description
Adc_GroupType	Group	input	- index of group .
	TriggerSource-	input	Trigger source to be use for the group. (Configuration file should contain the TriggerSource for called group).
Adc_GroupType	Group	input	Index of group .
Adc_HwTriggerTimerType	TriggerSource	input	Trigger source to be use for the group. (Configuration file should contain the TriggerSource for called group).

3.5.3.23 Function Adc_HwResultReadGroup

Function reads the result of the hardware triggered groups conversion result.

Details:

This non autosar API is used to read the result of the hardware triggered groups conversion result from the ADC hardware register in this case the CTU interrupt will be disabled for the group. The VALID bit CDR register will be cleared automatically when we read the results from the channel data register. If the user calls non-autosar function

Adc_HwResultReadGroup() once again before the next conversion takes place, the Adc_HwResultReadGroup() returns E_NOT_OK.

Note

The function Service ID[hex]: 0x0D. Violates Adc_NonASR_c_REF_3: External function could be made static.

Return: Std_ReturnType - E_OK or E_NOT_OK.

This non autosar API is used to read the result of the hardware triggered groups conversion result from the ADC hardware register in this case the CTU interrupt will be disabled for the group.

Note

The function Service ID[hex]: 0x0D.

Return: `Adc_ValueGroupType` pointer to internal buffer.

Prototype: `Std_ReturnType Adc_HwResultReadGroup(Adc_GroupType Group, Adc_ValueGroupType *DataPtr);`

Table 3-31. Adc_HwResultReadGroup Arguments

Type	Name	Direction	Description
<code>Adc_GroupType</code>	Group	input	Index of group .
<code>Adc_ValueGroupType *</code>	DataPtr	output	Pointer to a buffer which will be filled by the conversion results.
<code>Adc_GroupType</code>	Group	input	Index of group.
<code>Adc_ValueGroupType *</code>	DataPtr	output	Pointer to a buffer which will be filled by the conversion results.

3.5.3.24 Function ADC_SetClockMode

Set the clock ADC clock prescaler.

Details:

This function set the clock ADC clock prescaler

Note

The function NON Autosar Service. Synchronous. Non Reentrant function. Violates `Adc_NonASR_c_REF_3`: External function could be made static.

Prototype: `Std_ReturnType ADC_SetClockMode(Adc_NonASR_DualClockModeType ClockMode);`

Table 3-32. ADC_SetClockMode Arguments

Type	Name	Direction	Description
<code>Adc_NonASR_DualClockModeType</code>	ClockMode	input	NORMAL mode or ALTERNATE mode.

3.5.4 Structs Reference

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

3.5.4.1 Structure Adc_ConfigType

Structure for Configuration data.

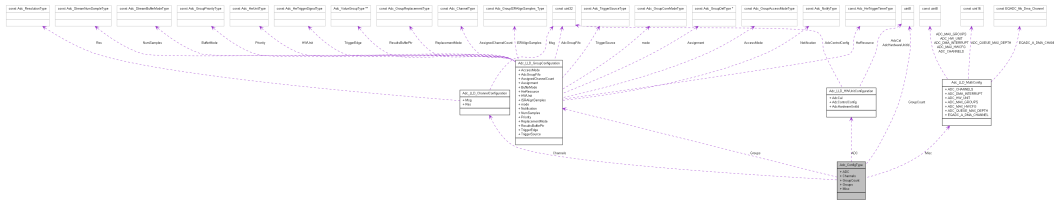


Figure 3-9. Struct Adc_ConfigType

Declaration:

```
typedef struct
{
    const Adc_LLD_HWUnitConfiguration * ADC,
    const Adc_LLD_ChannelConfiguration ** Channels,
    Adc_GroupType GroupCount,
    const Adc_LLD_GroupConfiguration * Groups,
    const Adc_LLD_MultiConfig Misc
} Adc_ConfigType;
```

Table 3-33. Structure Adc_ConfigType member description

Member	Description
ADC	
Channels	
GroupCount	
Groups	
Misc	

3.5.4.2 Structure Adc_GroupStatusType

Structure for group status.

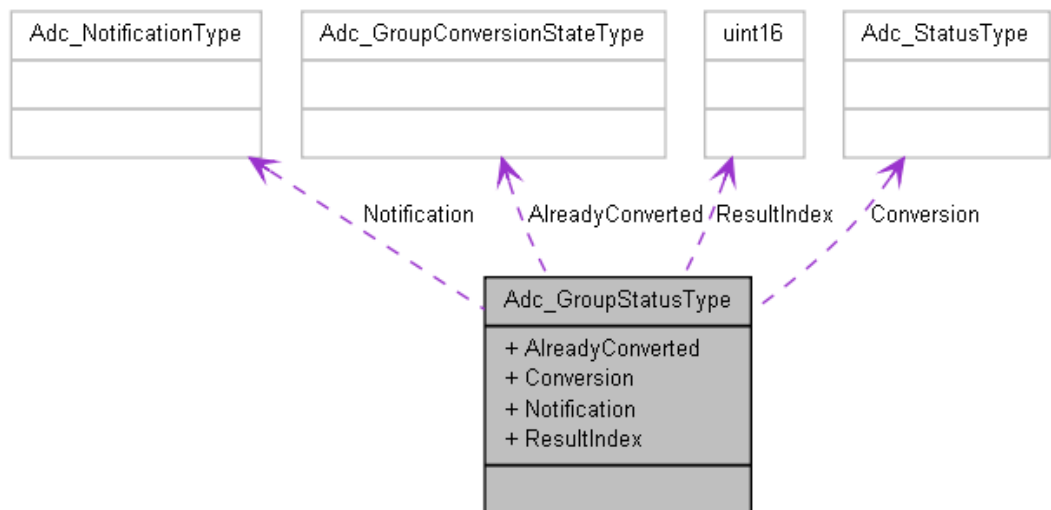


Figure 3-10. Struct Adc_GroupStatusType

Declaration:

```
typedef struct
{
    Adc_GroupConversionStateType AlreadyConverted,
    Adc_StatusType Conversion,
    Adc_NotificationType Notification,
    Adc_StreamNumSampleType ResultIndex
} Adc_GroupStatusType;
```

Table 3-34. Structure Adc_GroupStatusType member description

Member	Description
AlreadyConverted	
Conversion	
Notification	
ResultIndex	

3.5.4.3 Structure Adc_LLD_ChannelConfiguration

Adc_LLD_ChannelConfiguration. This type contains channel configuration details. It contains the information like:

- Fifo conversion command message
- Resolution of Adc channel.

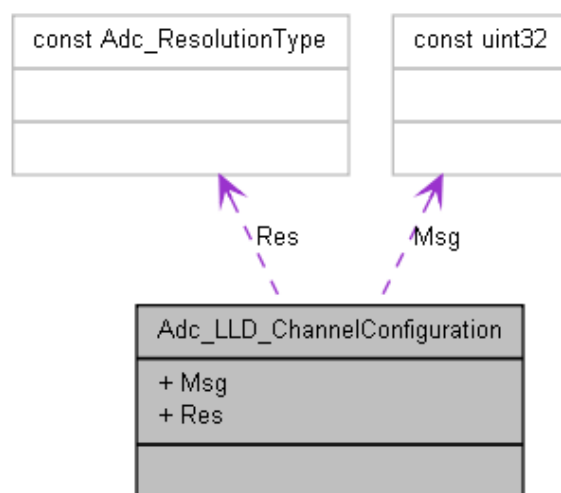


Figure 3-11. Struct Adc_LLD_ChannelConfiguration

Declaration:

```
typedef struct
{
    const uint32 Msg,
    const Adc_ResolutionType Res
} Adc_LLD_ChannelConfiguration;
```

Table 3-35. Structure `Adc_LLD_ChannelConfiguration` member description

Member	Description
Msg	Conversion command message.
Res	Resolution of the ADC hardware (8 or 10 or 12 bits).

3.5.4.4 Structure Adc_LLD_GroupConfiguration

Adc_LLDD_GroupConfiguration. This type contains Group configuration details. It contains the information like:

- Hardware unit id to which the group belongs to
- Access mode of group
- Notification function pointer
- Number of streaming samples for streaming mode
- Number of channels assigned to this group.



Figure 3-12. Struct Adc_LLD_GroupConfiguration

Declaration:

```
typedef struct
{
    const Adc_GroupAccessModeType AccessMode,
    const uint32 AdcGroupFifo,
    const Adc_ChannelType AssignedChannelCount,
    const Adc_GroupDefType * Assignment,
    const Adc_StreamBufferModeType BufferMode,
    const Adc_HwTriggerTimerType HwResource,
    const Adc_HwUnitType HWUnit,
    const Adc_GroupISRAAlignSamples_Type ISRAAlignSamples,
    const Adc_GroupConvModeType mode,
    const Adc_NotifyType Notification,
    const Adc_StreamNumSampleType NumSamples,
    const Adc_GroupPriorityType Priority,
    const Adc_GroupReplacementType ReplacementMode,
    Adc_ValueGroupType ** ResultsBufferPtr,
    const Adc_HwTriggerSignalType TriggerEdge,
    const Adc_TriggerSourceType TriggerSource
} Adc_LLD_GroupConfiguration;
```

Table 3-36. Structure Adc_LLD_GroupConfiguration member description

Member	Description
AccessMode	ADC_ACCESS_MODE_SINGLE, ADC_ACCESS_MODE_STREAMING.
AdcGroupFifo	Fifo used for this group.
AssignedChannelCount	Number of channels configured in the group.
Assignment	Pointer to assigned channels in the group.
BufferMode	ADC_STREAM_BUFFER_LINEAR, ADC_STREAM_BUFFER_CIRCULAR.
HwResource	Resource of the selected hw trigger.
HWUnit	Hw unit to which the group belongs to.
ISRAAlignSamples	Samples alignment.
mode	conversion mode (OneShot/Continuous).
Notification	Pointer to notification function.
NumSamples	Number of samples in streaming access mode.
Priority	Priority of the group.
ReplacementMode	Replacement mode (Abort restart or suspend resume) of group.
ResultsBufferPtr	Pointer to user defined result buffer.
TriggerEdge	Trigger Edge (Rising or falling) for hardware triggered groups.
TriggerSource	Trigger source (Hw/Sw trigger) of group.

3.5.4.5 Structure Adc_LLD_HWUnitConfiguration

Adc_LLD_HWUnitConfiguration. This type contains hardware configuration details. It contains the information like:

- Hardware unit id configured
- hardware control register details
- Calibration details.

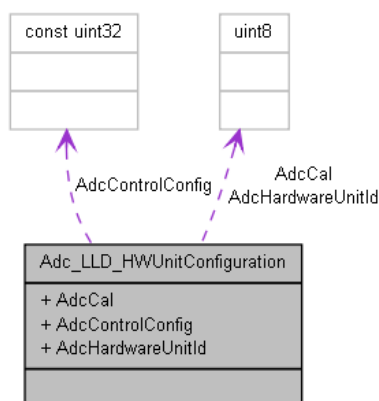


Figure 3-13. Struct Adc_LLD_HWUnitConfiguration

Declaration:

```

typedef struct
{
    uint8 AdcCal,
        const uint32 AdcControlConfig,
        const Adc_HwUnitType AdcHardwareUnitId
} Adc_LLD_HWUnitConfiguration;
  
```

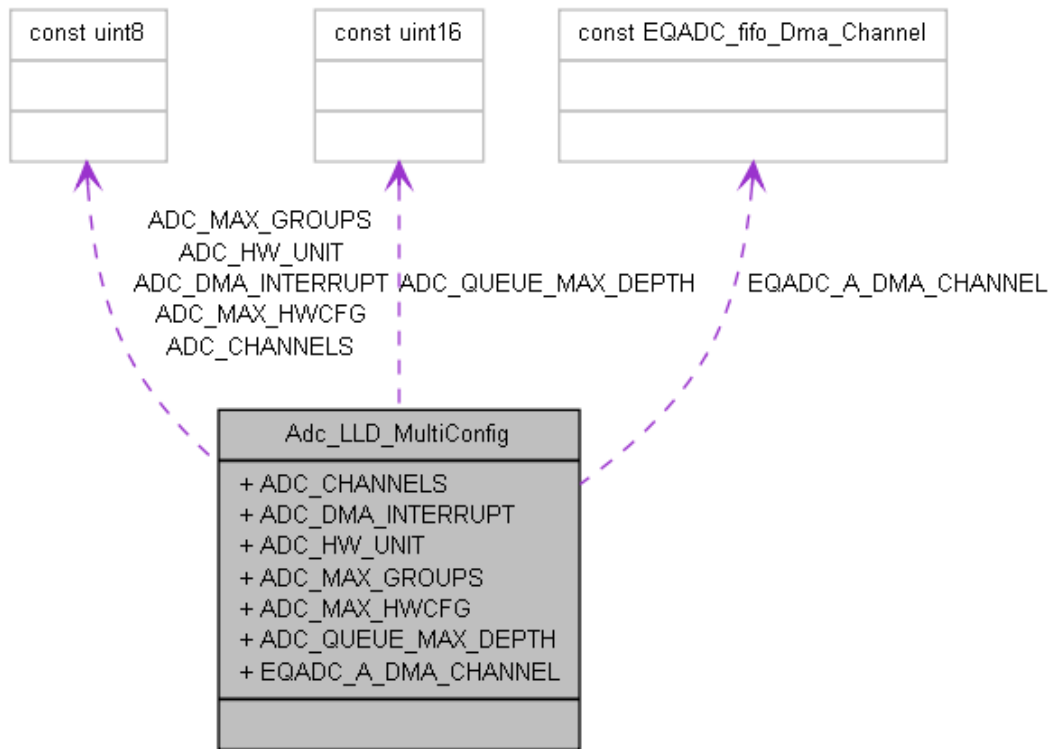
Table 3-37. Structure Adc_LLD_HWUnitConfiguration member description

Member	Description
AdcCal	ADCN_CR control register value.
AdcControlConfig	Configuration of ADC hardware control register.
AdcHardwareUnitId	Adc hardware unit id.

3.5.4.6 Structure Adc_LLD_MultiConfig

`Adc_LLD_MultiConfig`. This type contains General configuration details. It contains the information like:

- Configured mode of ADC i.e DMA or INTERRUPT
- Maximum groups configured in the configuration
- Maximum groups configured in the configuration
- Maximum depth of queue
- Maximum number of channels in hardware unit.

Figure 3-14. Struct `Adc_LLD_MultiConfig`**Declaration:**

```
typedef struct
{
    const uint8 ADC_CHANNELS[ADC_HW_MAX],
        const uint8 ADC_DMA_INTERRUPT,
        const uint8 ADC_HW_UNIT[ADC_HW_MAX],
        const uint8 ADC_MAX_GROUPS,
        const uint8 ADC_MAX_HWCFG,
        const uint16 ADC_QUEUE_MAX_DEPTH,
        const EQADC_fifo_Dma_Channel EQADC_A_DMA_CHANNEL
} Adc_LLD_MultiConfig;
```

Table 3-38. Structure `Adc_LLD_MultiConfig` member description

Member	Description
<code>ADC_CHANNELS</code>	Number of channels of the hw unit x.
<code>ADC_DMA_INTERRUPT</code>	Dma or interrupt driven.
<code>ADC_HW_UNIT</code>	Indicate if unit x is active (configured) STD_ON/STD_OFF.
<code>ADC_MAX_GROUPS</code>	The number of the maximum groups in the current configuration.
<code>ADC_MAX_HWCFG</code>	The number of the maximum number of ADC hardware units configured in the configset.
<code>ADC_QUEUE_MAX_DEPTH</code>	Depth of the queue.
<code>EQADC_A_DMA_CHANNEL</code>	If dma driven then indicates the dma channel number for HW UNITS 0 and 1.

3.5.4.7 Structure Adc_UnitStatusType

Structure for hardware unit status.

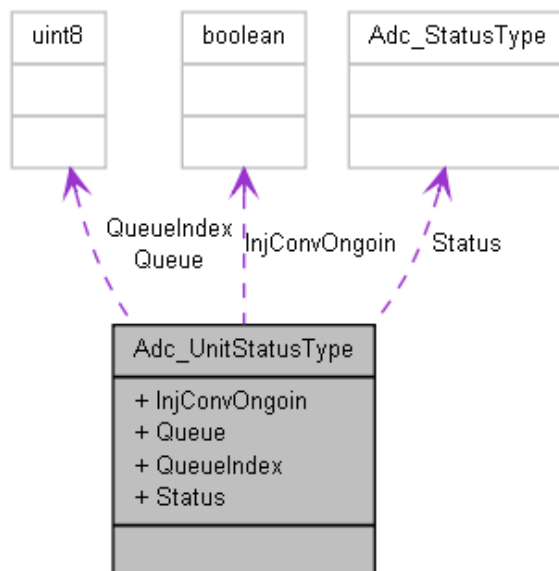


Figure 3-15. Struct Adc_UnitStatusType

Declaration:

```
typedef struct
{
    boolean InjConvOngoin,
    Adc_GroupType Queue[ADC_QUEUE_MAX_DEPTH_MAX],
    Adc_QueueIndexType QueueIndex,
    Adc_StatusType Status
} Adc_UnitStatusType;
```

Table 3-39. Structure Adc_UnitStatusType member description

Member	Description
InjConvOngoin	
Queue	
QueueIndex	
Status	

3.5.5 Types Reference

Types supported by the driver are as per AUTOSAR ShortName Driver software specification Version 3.0 .

3.5.5.1 Typedef Adc_ChannelType

Details:

Numeric ID of an ADC channel.

Type: uint8

Numeric ID of an ADC channel.

3.5.5.2 Typedef Adc_ClockSourceType

Details:

Clock source for ADC - not used on XPC560XP

Type: uint8

Clock source for ADC - not used on XPC560XP

3.5.5.3 Typedef Adc_ConversionTimeType

Details:

Conversion time

Type: uint16

Conversion time

3.5.5.4 Typedef Adc_GroupDefType

Details:

Definition of channels in a group

Type: Adc_ChannelType

Definition of channels in a group

3.5.5.5 Typedef Adc_GroupPriorityType

Details:

ADC Channel group priority

Type: uint8

ADC Channel group priority

3.5.5.6 Typedef Adc_GroupType

Details:

Numeric ID of an ADC channel group. No uC dependent

Type: uint8

Numeric ID of an ADC channel group. No uC dependent

3.5.5.7 Typedef Adc_HwTriggerTimerType

Details:

Type for the reload value of the ADC module embedded timer

Type: uint16

Type for the reload value of the ADC module embedded timer

3.5.5.8 Typedef Adc_HwUnitType

Details:

Numeric ID of an ADC Hw Unit

Type: uint8

Numeric ID of an ADC Hw Unit

3.5.5.9 Typedef Adc_NotifyType

Details:

Notification function pointer definition

Type: void(*)

Notification function pointer definition

3.5.5.10 Typedef Adc_PrescaleType

Details:

Clock prescaler factor

Type: uint8

Clock prescaler factor

3.5.5.11 Typedef Adc_QueueIndexType

Details:

Index for the queue of groups

Type: uint8

Index for the queue of groups

3.5.5.12 Typedef Adc_ResolutionType

Details:

Channel resolution in number of bits

Type: uint8

Channel resolution in number of bits

3.5.5.13 Typedef Adc_SamplingTimeType

Details:

Sampling time

Type: uint8

Sampling time

3.5.5.14 Typedef Adc_StreamNumSampleType

Details:

Number of samples of a streaming conversion buffer

Type: uint16

Number of samples of a streaming conversion buffer

3.5.5.15 Typedef Adc_ValueGroupType

Details:

Type for ADC result of one channel. 10bit for XPC560XP Platform

Type: uint16

Type for ADC result of one channel. 10bit for XPC560XP Platform

3.5.5.16 Typedef Adc_VoltageSourceType

Details:

Reference voltage source

Type: uint16

Reference voltage source

3.5.5.17 Typedef Adc_WdgNotifyType

Details:

Notification function pointer definition - WDG

Type: void(*)

Notification function pointer definition - WDG

3.5.5.18 Typedef EQADC_fifo_Dma_Channel

Details:

configured dma channels for the eQADC_A and eQADC_B

Type: uint8

configured dma channels for the eQADC_A and eQADC_B

3.5.6 Variables Reference

Variables supported by the driver are as per AUTOSAR ShortName Driver software specification Version 3.0 .

3.5.6.1 Variable Adc_Cfg_Ptr

Segment descriptor for data arranged in array.

Declaration:

```
const Adc_ConfigType* Adc_Cfg_Ptr
```

3.5.6.2 Variable ADC_GroupStatus

Array to hold the group status.

Declaration:

```
Adc_GroupStatusType ADC_GroupStatus[ADC_MAX_GROUPS]
```

3.5.6.3 Variable ADC_UnitStatus

Structure for the unit status.

Declaration:

```
Adc_UnitStatusType ADC_UnitStatus[ADC_MAX_HW_UNITS]
```

3.6 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 ShortName Driver. The most of the parameters are described below.

4.1 Configuration elements of Adc

Included forms :

- IMPLEMENTATION_CONFIG_VARIANT
- AdcGeneral
- AdcPublishedInformation
- CommonPublishedInformation
- NonAutosar
- AdcConfigSet

Table 4-1. Revision table

Revision	Date
1.0	2010-09-15T10:30:00

4.2 Form IMPLEMENTATION_CONFIG_VARIANT

Configuration classes Enable the parameters that are editable for specific configuration classes

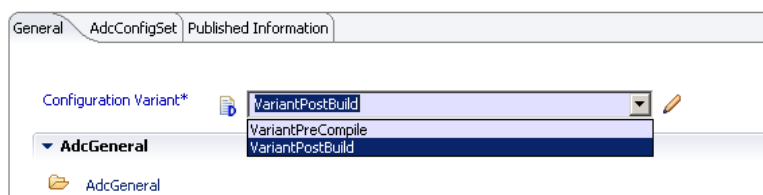


Figure 4-1. Tresos Plugin snapshot for IMPLEMENTATION_CONFIG_VARIANT form.

Table 4-2. Attribute IMPLEMENTATION_CONFIG_VARIANT detailed description

Property	Value
Label	Configuration Variant
Default	VariantPostBuild
Range	VariantPreCompile VariantPostBuild

4.3 Form AdcGeneral

General configuration (parameters) of the ADC Driver software module.

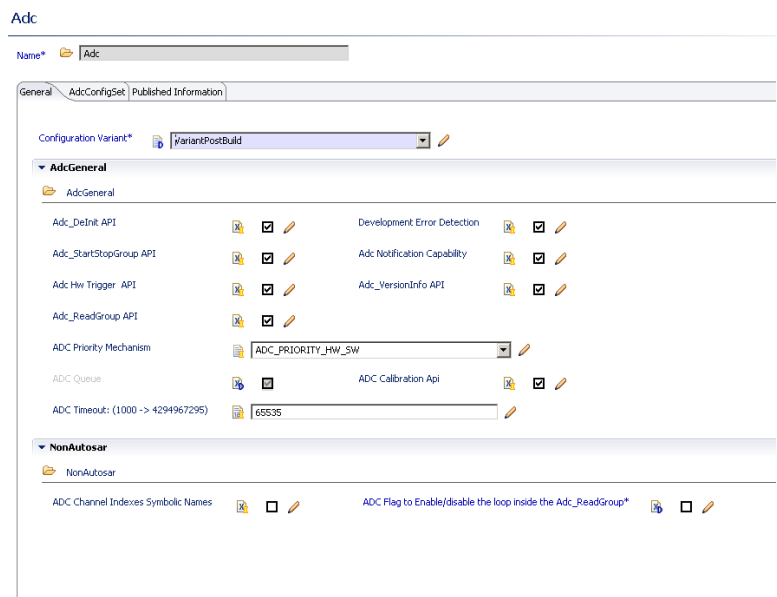


Figure 4-2. Tresos Plugin snapshot for AdcGeneral form.

4.3.1 AdcDeInitApi (AdcGeneral)

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

Table 4-3. Attribute AdcDeInitApi (AdcGeneral) detailed description

Property	Value
Label	Adc_DeInit API
Type	BOOLEAN
Origin	AUTOSAR_ECUC

Table continues on the next page...

Table 4-3. Attribute AdcDelnitApi (AdcGeneral) detailed description (continued)

Property	Value
Symbolic Name	false
Default	true
Lower Multiplicity	1
Upper Multiplicity	1

4.3.2 AdcDevErrorDetect (AdcGeneral)

Enable/Disable Development Error Detection

Table 4-4. Attribute AdcDevErrorDetect (AdcGeneral) detailed description

Property	Value
Label	Development Error Detection
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true
Lower Multiplicity	1
Upper Multiplicity	1

4.3.3 AdcEnableStartStopGroupApi (AdcGeneral)

Enable/Disable the services Adc_StartGroupConversion() and Adc_StopGroupConversion from the code.

Table 4-5. Attribute AdcEnableStartStopGroupApi (AdcGeneral) detailed description

Property	Value
Label	Adc_StartStopGroup API
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true
Lower Multiplicity	1
Upper Multiplicity	1

4.3.4 AdcGrpNotifCapability (AdcGeneral)

Determines, if the group notification mechanism is available at runtime.

Table 4-6. Attribute AdcGrpNotifCapability (AdcGeneral) detailed description

Property	Value
Label	Adc Notification Capability
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true
Lower Multiplicity	1
Upper Multiplicity	1

4.3.5 AdcHwTriggerApi (AdcGeneral)

Adds / removes the services Adc_EnableHardwareTrigger() and Adc_DisableHardwareTrigger() from the code.

Table 4-7. Attribute AdcHwTriggerApi (AdcGeneral) detailed description

Property	Value
Label	Adc Hw Trigger API
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	false
Lower Multiplicity	1
Upper Multiplicity	1

4.3.6 AdcVersionInfoApi (AdcGeneral)

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

Table 4-8. Attribute AdcVersionInfoApi (AdcGeneral) detailed description

Property	Value
Label	Adc_VersionInfo API
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true
Lower Multiplicity	1
Upper Multiplicity	1

4.3.7 AdcReadGroupApi (AdcGeneral)

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

Table 4-9. Attribute AdcReadGroupApi (AdcGeneral) detailed description

Property	Value
Label	Adc_ReadGroup API
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true
Lower Multiplicity	1
Upper Multiplicity	1

4.3.8 AdcPriorityImplementation (AdcGeneral)

Select the Priority mechanism.

Table 4-10. Attribute AdcPriorityImplementation (AdcGeneral) detailed description

Property	Value
Label	ADC Priority Mechanism
Type	ENUMERATION
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	ADC_PRIORITY_NONE

Table continues on the next page...

Table 4-10. Attribute AdcPriorityImplementation (AdcGeneral) detailed description (continued)

Property	Value
Lower Multiplicity	1
Upper Multiplicity	1
Range	ADC_PRIORITY_HW ADC_PRIORITY_HW_SW ADC_PRIORITY_NONE

4.3.9 AdcEnableQueuing (AdcGeneral)

Enable/Disable the Queue. Note that if AdcPriorityImplementation=ADC_PRIORITY_HW_SW this field is always enabled.

Table 4-11. Attribute AdcEnableQueuing (AdcGeneral) detailed description

Property	Value
Label	ADC Queue
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true
Lower Multiplicity	1
Upper Multiplicity	1

4.3.10 AdcCalibrationApi (AdcGeneral)

Adds/removes (STD_ON/STD_OFF) the calibration code from the driver. Note that besides this configuration item you also need to enable calibration on Hw unit (Adc/AdcConfigSet/AdcHwUnit/AdcCalibration) and on each channel that should be calibrated (Adc/AdcConfigSet/AdcHwUnit/AdcChannel/AdcChannelCalibration). Note: This is an Implementation Specific Parameter.

Table 4-12. Attribute AdcCalibrationApi (AdcGeneral) detailed description

Property	Value
Label	ADC Calibration Api
Origin	Custom
Symbolic Name	false

Table continues on the next page...

Table 4-12. Attribute AdcCalibrationApi (AdcGeneral) detailed description (continued)

Property	Value
Default	true
Lower Multiplicity	1
Upper Multiplicity	1

4.3.11 AdcTimeout (AdcGeneral)

This timeout is used to wait until the FIFO status is updated when EOQ flag is set or Result fifo is cleared or fifo is disabled. If the Status is not updated then after this timeout the ADC_E_TIMEOUT production error will be reported and the rest of the functionality will be skipped.

Table 4-13. Attribute AdcTimeout (AdcGeneral) detailed description

Property	Value
Label	ADC Timeout:
Origin	Custom
Symbolic Name	false
Default	65535
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range <=4294967295 >=1000

4.4 Form AdcPublishedInformation

Additional published parameters not covered by CommonPublishedInformation container. Note that these parameters do not have any configuration class setting, since they are published information.

▼ AdcPublishedInformation

AdcPublishedInformation

ADC Channel Value Signed False

ADC Group First Channel Fixed False

ADC Max Channel Resolution 12

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

4.4.1 AdcChannelValueSigned (AdcPublishedInformation)

Information whether the result value of the ADC driver has sign information (true) or not (false).

Table 4-14. Attribute AdcChannelValueSigned (AdcPublishedInformation) detailed description

Property	Value
Label	ADC Channel Value Signed
Type	BOOLEAN_LABEL
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	false
Lower Multiplicity	1
Upper Multiplicity	1

4.4.2 AdcGroupFirstChannelFixed (AdcPublishedInformation)

Information whether the first channel of an ADC Channel group can be configured (false) or is fixed (true) to a value determined by the ADC HW Unit.

Table 4-15. Attribute AdcGroupFirstChannelFixed (AdcPublishedInformation) detailed description

Property	Value
Label	ADC Group First Channel Fixed
Type	BOOLEAN_LABEL
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	false
Lower Multiplicity	1
Upper Multiplicity	1

4.4.3 AdcMaxChannelResolution (AdcPublishedInformation)

Maximum Channel resolution in bits (does not specify accuracy)

Table 4-16. Attribute AdcMaxChannelResolution (AdcPublishedInformation) detailed description

Property	Value
Label	ADC Max Channel Resolution
Type	INTEGER_LABEL
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	12
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range <div>>=0</div> <div><=255</div>

4.5 Form CommonPublishedInformation

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

Included forms :

▼ CommonPublishedInformation

CommonPublishedInformation

AUTOSAR Major Version 3

AUTOSAR Minor Version 0

AUTOSAR Patch Version 2

AUTOSAR Module Id 123

Software Major Version 2

Software Minor Version 0

Software Patch Version 0

Vendor Specific Name

Vendor ID 27

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

4.5.1 ArMajorVersion (CommonPublishedInformation)

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

Table 4-17. Attribute ArMajorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	AUTOSAR Major Version
Origin	Custom
Symbolic Name	false
Default	3
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=3 <=3

4.5.2 ArMinorVersion (CommonPublishedInformation)

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

Table 4-18. Attribute ArMinorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	AUTOSAR Minor Version
Origin	Custom
Symbolic Name	false
Default	0
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=0 <=0

4.5.3 ArPatchVersion (CommonPublishedInformation)

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

Table 4-19. Attribute ArPatchVersion (CommonPublishedInformation) detailed description

Property	Value
Label	AUTOSAR Patch Version
Origin	Custom
Symbolic Name	false
Default	2
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range ≥ 2 ≤ 2

4.5.4 ModuleId (CommonPublishedInformation)

Module ID of this module from Module List.

Table 4-20. Attribute ModuleId (CommonPublishedInformation) detailed description

Property	Value
Label	AUTOSAR Module Id
Origin	Custom
Symbolic Name	false
Default	123
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range ≥ 123 ≤ 123

4.5.5 SwMajorVersion (CommonPublishedInformation)

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

Table 4-21. Attribute SwMajorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	Software Major Version
Origin	Custom
Symbolic Name	false
Default	2
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range ≥ 2 ≤ 2

4.5.6 SwMinorVersion (CommonPublishedInformation)

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

Table 4-22. Attribute SwMinorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	Software Minor Version
Origin	Custom
Symbolic Name	false
Default	0
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range ≥ 0 ≤ 0

4.5.7 SwPatchVersion (CommonPublishedInformation)

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

Table 4-23. Attribute SwPatchVersion (CommonPublishedInformation) detailed description

Property	Value
Label	Software Patch Version
Origin	Custom
Symbolic Name	false
Default	0
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range <div style="margin-left: 20px;">>=0</div> <div style="margin-left: 20px;"><=0</div>

4.5.8 VendorId (CommonPublishedInformation)

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

Table 4-24. Attribute VendorId (CommonPublishedInformation) detailed description

Property	Value
Label	Vendor ID
Origin	Custom
Symbolic Name	false
Default	27
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range <div style="margin-left: 20px;">>=27</div> <div style="margin-left: 20px;"><=27</div>

4.5.9 VendorApiInfix (VendorApiInfix)

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

<ModuleName>_>VendorId>_<VendorApiInfix><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 4-25. Attribute VendorApiInfix (VendorApiInfix) detailed description

Property	Value
Label	Vendor Specific Name
Origin	AUTOSAR_ECUC V1.0.0
Symbolic Name	false
Lower Multiplicity	1
Upper Multiplicity	1

4.6 Form NonAutosar

General configuration (parameters) for Non-Autosar.



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

4.6.1 AdcChIndexSymNames (NonAutosar)

This is used to generate ADC symbolic names, that depend also on the ADC group to which each ADC channel is mapped. The generated symbolic name will be something like: #define "ADC_GroupName"_"ADC_ChannelName" "Channel index value", where "Channel index value" is the channel index in the current group. Channel indexes in each group are generated to allow result buffer access by symbolic names.

Table 4-26. Attribute AdcChIndexSymNames (NonAutosar) detailed description

Property	Value
Label	ADC Channel Indexes Symbolic Names
Type	BOOLEAN
Origin	Custom
Symbolic Name	false

Table continues on the next page...

Table 4-26. Attribute AdcChIndexSymNames (NonAutosar) detailed description (continued)

Property	Value
Default	false
Lower Multiplicity	1
Upper Multiplicity	1

4.6.2 AdcReadGroupOptimization (NonAutosar)

Enable/Disable the Adc_ReadGroup() API optimization. It means enable\disable the conversion values copy loop, inside the Adc_ReadGroup() API, on all the channel inside a group. The group MUST have a stream depth limited to 1. Moreover the Adc_SetupResultBuffer() and Adc_ReadGroup() MUST be called using the "same" buffer pointer.

Table 4-27. Attribute AdcReadGroupOptimization (NonAutosar) detailed description

Property	Value
Label	ADC Flag to Enable/disable the loop inside the Adc_ReadGroup
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false
Lower Multiplicity	1
Upper Multiplicity	1

4.7 Form AdcConfigSet

This is the base container that contains the post-build selectable configuration parameters

Included forms :

- [Form AdcGeneric](#)
- [Form AdcHwUnit](#)

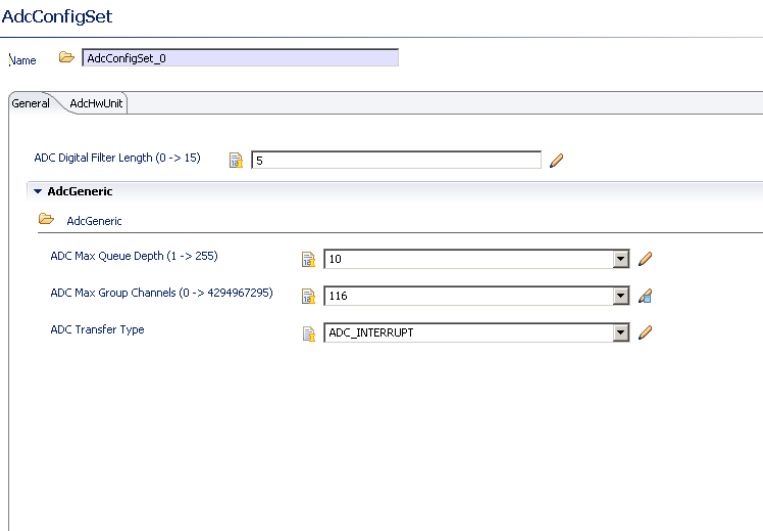


Figure 4-6. TRESOS Plugin snapshot for AdcConfigSet form.

4.7.1 AdcDigitalFilterLength (AdcConfigSet)

Digital Filter Length (DFL bits in EQADC_ETDFR register). Specifies the minimum number of system clocks that must the digital filter counter count to recognize a logic state change on EQADC trigger input. DFL ----- Minimum number of system clocks 0 ----- 2; 1 ----- 3; 2 ----- 5; 3 ----- 9; 4 ----- 17; 5 ----- 33; 6 ----- 65; 7 ----- 129; 8 ----- 257; 9 ----- 513; 10 ----- 1025; 11 ----- 2049; 12 ----- 4097; 13 ----- 8193; 14 ----- 16385; 15 ----- 32769. Note: This is an Implementation Specific Parameter.

Table 4-28. Attribute AdcDigitalFilterLength (AdcConfigSet) detailed description

Property	Value
Label	ADC Digital Filter Length
Origin	Custom
Symbolic Name	false
Default	5
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range <=15 >=0

4.7.2 Form AdcGeneric

General configuration parameters for the current configuration of the ADC Driver software module.

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

Figure 4-7. Tressos Plugin snapshot for AdcGeneric form.

4.7.2.1 AdcPriorityQueueMaxDepth (AdcGeneric)

Maximum depth of queue used for queuing of incoming conversion requests when hardware unit is busy. Note: This is an Implementation Specific Parameter.

Table 4-29. Attribute AdcPriorityQueueMaxDepth (AdcGeneric) detailed description

Property	Value
Label	ADC Max Queue Depth
Origin	Custom
Symbolic Name	false
Default	1
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range <div> <div><=255</div> <div>>=1</div> </div>

4.7.2.2 AdcMaxGroupChannels (AdcGeneric)

Maximum number of ADC HW channels configurable and allowed for a group: it's the maximum number of channels available for that package.

Table 4-30. Attribute AdcMaxGroupChannels (AdcGeneric) detailed description

Property	Value
Label	ADC Max Group Channels
Origin	Custom
Symbolic Name	false
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=0 <=4294967295

4.7.2.3 AdcTransferType (AdcGeneric)

Select the Interrupt or Dma transfer Type. Note: This is an Implementation Specific Parameter.

Table 4-31. Attribute AdcTransferType (AdcGeneric) detailed description

Property	Value
Label	ADC Transfer Type
Origin	Custom
Symbolic Name	false
Default	ADC_INTERRUPT
Lower Multiplicity	1
Upper Multiplicity	1
Range	ADC_DMA ADC_INTERRUPT

4.7.3 Form AdcHwUnit

This container contains configuration of the hw unit.

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

Included forms :

- [Form AdcChannel](#)
- [Form AdcGroup](#)

Figure 4-8. Tressos Plugin snapshot for AdcHwUnit form.

4.7.3.1 AdcClockSource (AdcHwUnit)

This parameter is used to select the system clock signal or the prescaler output clock signal. 0 = Prescaler output clock is selected. 1 = System clock is selected - maximum frequency.

Table 4-32. Attribute AdcClockSource (AdcHwUnit) detailed description

Property	Value
Label	ADC Source Clock
Type	INTEGER
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	0
Lower Multiplicity	0
Upper Multiplicity	1
Enable	true
Invalid	Range <div>>=0</div> <div><=1</div>

4.7.3.2 AdcHwUnitId (AdcHwUnit)

Specifies the used ADC Hardware Unit. eQADC (AdcHwUnitId is equals to 0 or 1, 0 means unit0 and 1 means unit1)

Table 4-33. Attribute AdcHwUnitId (AdcHwUnit) detailed description

Property	Value
Label	ADC Hardware Unit
Type	INTEGER
Origin	AUTOSAR_ECUC
Symbolic Name	true
Lower Multiplicity	1
Upper Multiplicity	1
Enable	true
Invalid	Range >=0 <=1

4.7.3.3 AdcPrescale (AdcHwUnit)

ADCn Clock Prescaler (ADCn_CLK_PS bits in ADCn_CR register). The ADCn_CLK_PS field controls the system clock divide factor for the ADCn clock as in the following table: ADCn_CLK_PS System Clock Divide Factor 0 ----- 2; 1 ----- 4; 2 ----- 6; ...; 30 ----- 62; 31 ----- 64; Note: This parameter value is applicable only when the Prescaler output clock is selected in AdcClockSource parameter

Table 4-34. Attribute AdcPrescale (AdcHwUnit) detailed description

Property	Value
Label	ADC Prescaler Value
Type	INTEGER
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	31
Lower Multiplicity	0
Upper Multiplicity	1
Enable	true
Invalid	Range <=31 >=0

4.7.3.4 AdcCalibration (AdcHwUnit)

Enables/disables the calibration on this hw unit. Besides this configuration parameter it is also needed to enable "AdcCalibrationApi" parameter and "AdcChannelCalibration" for the channel which should be calibrated. Note: This is an Implementation Specific Parameter.

Table 4-35. Attribute AdcCalibration (AdcHwUnit) detailed description

Property	Value
Label	ADC Hardware Calibration
Origin	Custom
Symbolic Name	false
Default	true
Lower Multiplicity	1
Upper Multiplicity	1

4.7.3.5 AdcExternalMultiplexing (AdcHwUnit)

Enables external multiplexing (ADCn_EMUX bit in ADCn_CR register). This parameter can not be enabled for both the hardware of eQADC_A or eQADC_B at the same time. Note: This is an Implementation Specific Parameter.

Table 4-36. Attribute AdcExternalMultiplexing (AdcHwUnit) detailed description

Property	Value
Label	ADC External Multiplexer
Origin	Custom
Symbolic Name	false
Default	false
Lower Multiplicity	1
Upper Multiplicity	1

4.7.3.6 Form AdcChannel

This container contains the channel configuration parameters.

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

Figure 4-9. TRESOS Plugin snapshot for AdcChannel form.

4.7.3.6.1 AdcChannelConvTime (AdcChannel)

This parameter is not used by the current implementation.

Table 4-37. Attribute AdcChannelConvTime (AdcChannel) detailed description

Property	Value
Label	ADC Channel Conversion Time
Type	INTEGER
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	0
Lower Multiplicity	0
Upper Multiplicity	1
Enable	false
Invalid	Range <div style="margin-left: 20px;">>=0</div> <div style="margin-left: 20px;"><=0</div>

4.7.3.6.2 AdcChannelId (AdcChannel)

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

Table 4-38. Attribute AdcChannelId (AdcChannel) detailed description

Property	Value
Label	ADC Channel ID
Type	INTEGER
Origin	AUTOSAR_ECUC
Symbolic Name	true
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=0 <=120

4.7.3.6.3 AdcHwChannel (AdcChannel)

Selects the physical Hardware Adc Channel. AN0_AN39 are dedicated physical channels for the eQADC ANW_64 to ANW_71, ANX_72 to ANX_79, ANY_80 to ANY_87, ANZ_88 to ANZ_95, ANR_224 to ANR_231, ANS_232 to ANS_239, ANT_240 to ANT_247 can be configured only when the AdcExternalMultiplexing parameter is true. Note: This is an Implementation Specific Parameter.

Table 4-39. Attribute AdcHwChannel (AdcChannel) detailed description

Property	Value
Label	Physical Hw Channel
Origin	Custom
Symbolic Name	false
Lower Multiplicity	1
Upper Multiplicity	1

4.7.3.6.4 AdcChannelRefVoltsrcHigh (AdcChannel)

Upper reference voltage source for each channel (optional).

Table 4-40. Attribute AdcChannelRefVoltsrcHigh (AdcChannel) detailed description

Property	Value
Label	High Reference Voltage
Type	INTEGER
Origin	AUTOSAR_ECUC
Symbolic Name	false

Table continues on the next page...

Table 4-40. Attribute AdcChannelRefVoltsrcHigh (AdcChannel) detailed description (continued)

Property	Value
Default	0
Lower Multiplicity	0
Upper Multiplicity	1
Enable	false
Invalid	Range >=0 <=0

4.7.3.6.5 AdcChannelRefVoltsrcLow (AdcChannel)

Lower reference voltage source for each channel (optional).

Table 4-41. Attribute AdcChannelRefVoltsrcLow (AdcChannel) detailed description

Property	Value
Label	Low Reference Voltage
Type	INTEGER
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	0
Lower Multiplicity	0
Upper Multiplicity	1
Enable	false
Invalid	Range >=0 <=0

4.7.3.6.6 AdcChannelResolution (AdcChannel)

Resolution of converted value. Valid resolution is 8, 10 or 12Bits.

Table 4-42. Attribute AdcChannelResolution (AdcChannel) detailed description

Property	Value
Label	ADC Channel Resolution
Type	INTEGER
Origin	AUTOSAR_ECUC

Table continues on the next page...

Table 4-42. Attribute AdcChannelResolution (AdcChannel) detailed description (continued)

Property	Value
Symbolic Name	false
Default	12
Lower Multiplicity	0
Upper Multiplicity	1
Enable	true
Invalid	Range >=8 <=12

4.7.3.6.7 AdcChannelCalibration (AdcChannel)

Indicates if the returning channel conversion result is calibrated (CAL bit in conversion command message). Note that besides this configuration item it is also needed to globally enable calibration by "AdcCalibrationApi" and to enable calibration on hw unit by "AdcCalibration" parameter. Note: This is an Implementation Specific Parameter.

Table 4-43. Attribute AdcChannelCalibration (AdcChannel) detailed description

Property	Value
Label	ADC Channel Calibration
Origin	Custom
Symbolic Name	false
Default	true
Lower Multiplicity	1
Upper Multiplicity	1
Enable	true

4.7.3.6.8 AdcChannelSampTime (AdcChannel)

Select the Sampling time in ADC clock cycles When AdcExternalMultiplexing parameter is enabled in this case ADC_64_CLOCK_CYCLES or ADC_128_CLOCK_CYCLES sampling cycles is recommended Note: This is an Implementation Specific Parameter.

Table 4-44. Attribute AdcChannelSampTime (AdcChannel) detailed description

Property	Value
Label	ADC Channel Sampling Time:

Table continues on the next page...

Table 4-44. Attribute AdcChannelSampTime (AdcChannel) detailed description (continued)

Property	Value
Type	INTEGER
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	3
Lower Multiplicity	0
Upper Multiplicity	1
Invalid	Range ≥ 0 ≤ 3

4.7.3.7 Form AdcGroup

This container contains the Group configuration parameters.

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

Included forms :

The screenshot shows the 'AdcGroup' configuration window. At the top, the 'Name' field is set to 'AdcGroup_0'. Below this, the 'General' tab is selected, showing a list of configuration parameters for 'AdcGroupDefinition'. The parameters are as follows:

- Group Access Mode: ADC_ACCESS_MODE_SINGLE
- Group Conversion Mode: ADC_CONV_MODE_ONESHOT
- Group Id (0 -> 255): 0
- Group Priority (0 -> 255): 10
- Group Replacement: ADC_GROUP_REPL_ABORT_RESTART
- Group Trigger Source: ADC_TRIGG_SRC_SW
- Group Trigger Signal: ADC_HW_TRIG_RISING_EDGE
- Group Trigger Timer (0 -> 0): 0
- Group Notification: Notification_0
- Group Streaming Buffer Mode: ADC_STREAM_BUFFER_LINEAR
- Group Buffer Pointer: ResultBuffer0
- Group Streaming Number Samples (1 -> 65535): 1
- ADC HW Trigger Sources: PIT0
- ADC Group FIFO: ADC_FIFO_0
- ISR Alignment Samples: ☒

Figure 4-10. TRESOS Plugin snapshot for AdcGroup form.

4.7.3.7.1 AdcGroupAccessMode (AdcGroup)

Type of access mode to group conversion results.

Table 4-45. Attribute AdcGroupAccessMode (AdcGroup) detailed description

Property	Value
Label	Group Access Mode
Type	ENUMERATION
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	ADC_ACCESS_MODE_SINGLE
Lower Multiplicity	1
Upper Multiplicity	1
Range	ADC_ACCESS_MODE_SINGLE ADC_ACCESS_MODE_STREAMING

4.7.3.7.2 AdcGroupConversionMode (AdcGroup)

Conversion mode of the channel group.

Table 4-46. Attribute AdcGroupConversionMode (AdcGroup) detailed description

Property	Value
Label	Group Conversion Mode
Type	ENUMERATION
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	ADC_CONV_MODE_ONESHOT
Lower Multiplicity	1
Upper Multiplicity	1
Range	ADC_CONV_MODE_ONESHOT ADC_CONV_MODE_CONTINUOUS

4.7.3.7.3 AdcGroupId (AdcGroup)

Group Id of the ADC group.

Table 4-47. Attribute AdcGroupId (AdcGroup) detailed description

Property	Value
Label	Group Id
Type	INTEGER
Origin	AUTOSAR_ECUC
Symbolic Name	true
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=0 <=255

4.7.3.7.4 AdcGroupPriority (AdcGroup)

Priority level of the group. This item is ignored if Adc/AdcGeneral/AdcPriorityImplementation is defined to ADC_PRIORITY_NONE

Table 4-48. Attribute AdcGroupPriority (AdcGroup) detailed description

Property	Value
Label	Group Priority
Type	INTEGER
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	0
Lower Multiplicity	1
Upper Multiplicity	1
Enable	true
Invalid	Range <=255 >=0

4.7.3.7.5 AdcGroupReplacement (AdcGroup)

Replacement mechanism used on ADC group level. It's fixed to Abort/Restart

Table 4-49. Attribute AdcGroupReplacement (AdcGroup) detailed description

Property	Value
Label	Group Replacement

Table continues on the next page...

Table 4-49. Attribute AdcGroupReplacement (AdcGroup) detailed description (continued)

Property	Value
Type	ENUMERATION
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	ADC_GROUP_REPL_ABORT_RESTART
Lower Multiplicity	1
Upper Multiplicity	1
Enable	false
Range	ADC_GROUP_REPL_ABORT_RESTART ADC_GROUP_REPL_SUSPEND_RESUME

4.7.3.7.6 AdcGroupTriggSrc (AdcGroup)

Select the Trigger Source for ADC Unit. It's possible select Hw or Sw trigger.

Table 4-50. Attribute AdcGroupTriggSrc (AdcGroup) detailed description

Property	Value
Label	Group Trigger Source
Type	ENUMERATION
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	ADC_TRIGG_SRC_SW
Lower Multiplicity	1
Upper Multiplicity	1
Range	ADC_TRIGG_SRC_HW ADC_TRIGG_SRC_SW

4.7.3.7.7 AdcHwTrigSignal (AdcGroup)

Configures on which edge of the hardware trigger signal the driver should reach.

Table 4-51. Attribute AdcHwTrigSignal (AdcGroup) detailed description

Property	Value
Label	Group Trigger Signal
Type	ENUMERATION
Origin	AUTOSAR_ECUC
Symbolic Name	false

Table continues on the next page...

Table 4-51. Attribute AdcHwTrigSignal (AdcGroup) detailed description (continued)

Property	Value
Default	ADC_HW_TRIG_RISING_EDGE
Lower Multiplicity	1
Upper Multiplicity	1
Range	ADC_HW_TRIG_FALLING_EDGE ADC_HW_TRIG_RISING_EDGE

4.7.3.7.8 AdcHwTrigTimer (AdcGroup)

Isn't used on this version.

Table 4-52. Attribute AdcHwTrigTimer (AdcGroup) detailed description

Property	Value
Label	Group Trigger Timer
Type	INTEGER
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	0
Lower Multiplicity	0
Upper Multiplicity	1
Enable	false
Invalid	Range >=0 <=0

4.7.3.7.9 AdcNotification (AdcGroup)

This function pointer is called everytime when the conversion of this group is completed.

Table 4-53. Attribute AdcNotification (AdcGroup) detailed description

Property	Value
Label	Group Notification
Type	FUNCTION-NAME
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	NULL_PTR
Lower Multiplicity	1
Upper Multiplicity	1

4.7.3.7.10 AdcStreamingBufferMode (AdcGroup)

Select the Streaming mode.If the buffer is linear, when it is full the conversion is stopped.

Table 4-54. Attribute AdcStreamingBufferMode (AdcGroup) detailed description

Property	Value
Label	Group Streaming Buffer Mode
Type	ENUMERATION
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	ADC_STREAM_BUFFER_LINEAR
Lower Multiplicity	1
Upper Multiplicity	1
Range	ADC_STREAM_BUFFER_CIRCULAR ADC_STREAM_BUFFER_LINEAR

4.7.3.7.11 AdcResultBufferPointer (AdcGroup)

Pointer to the Data Buffer (destination for conversion results).

Table 4-55. Attribute AdcResultBufferPointer (AdcGroup) detailed description

Property	Value
Label	Group Buffer Pointer
Type	LINKER-SYMBOL
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	AdcResultBufferPointer
Lower Multiplicity	1
Upper Multiplicity	1

4.7.3.7.12 AdcStreamingNumSamples (AdcGroup)

Number of samples to be acquired per channel.

Table 4-56. Attribute AdcStreamingNumSamples (AdcGroup) detailed description

Property	Value
Label	Group Streaming Number Samples
Type	INTEGER
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	1
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=1 <=65535

4.7.3.7.13 AdcHwTrigSrc (AdcGroup)

Selects the source for the eQADC trigger inputs (bits eTSEL0..5 in SIU_ISEL3 register). Supported Trigger Sources are:

ETRIG0,ETRIG1,RTI,PIT0,PIT1,PIT2,PIT3,ETRIG2,ETRIG3,eTPU30_PIT0,eTPU30_PIT1,eTPU31_PIT0,
eTPU31_PIT1,eTPU28,eTPU29,eTPU30,eTPU31,eMIOS10_PIT2,eMIOS10_PIT3,eMIOS11_PIT2, eMIOS11_PIT3,eMIOS23. ETRIG0, ETRIG3 triggers support CFIFO0, CFIFO2 and CFIFO4, ETRIG1, ETRIG2 triggers support CFIFO1, CFIFO3 and CFIFO5. Note: This is an Implementation Specific Parameter.

Table 4-57. Attribute AdcHwTrigSrc (AdcGroup) detailed description

Property	Value
Label	ADC HW Trigger Sources
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	PIT0
Lower Multiplicity	1
Upper Multiplicity	1

4.7.3.7.14 AdcGroupFifo (AdcGroup)

Selects the FIFO on which the channels belongs to the group converted. ADC_FIFO_0 to ADC_FIFO_2 is used for unit0 while ADC_FIFO_3 to ADC_FIFO_5 is used for unit1 by eQADC Note: This is an Implementation Specific Parameter.

Table 4-58. Attribute AdcGroupFifo (AdcGroup) detailed description

Property	Value
Label	ADC Group FIFO
Origin	Custom
Symbolic Name	false
Default	ADC_FIFO_0
Lower Multiplicity	1
Upper Multiplicity	1
Enable	true
Range	ADC_FIFO_0 ADC_FIFO_1 ADC_FIFO_2 ADC_FIFO_3 ADC_FIFO_4 ADC_FIFO_5

4.7.3.7.15 AdcGroupISRAlignSamples (AdcGroup)

Enable/Disable right alignment of each sample inside ISR.

Table 4-59. Attribute AdcGroupISRAlignSamples (AdcGroup) detailed description

Property	Value
Label	ISR Alignment Samples
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true
Lower Multiplicity	1
Upper Multiplicity	1

4.7.3.7.16 AdcGroupDefinition (AdcGroupDefinition)

Assignment of channels to a channel group. For each AdcChannel that should belong to the group, a reference needs to be defined.

Table 4-60. Attribute AdcGroupDefinition (AdcGroupDefinition) detailed description

Property	Value
Type	REFERENCE

Table continues on the next page...

Table 4-60. Attribute AdcGroupDefinition (AdcGroupDefinition) detailed description (continued)

Property	Value
Origin	AUTOSAR_ECUC

How to Reach Us:

Home Page:

www.freescale.com

Web Support:

<http://www.freescale.com/support>

USA/Europe or Locations Not Listed:

Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:

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

Japan:

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

Asia/Pacific:

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

For Literature Requests Only:

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

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductors products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor 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. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claims alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics as their non-RoHS-complaint and/or non-Pb-free counterparts. For further information, see <http://www.freescale.com> or contact your Freescale sales representative.

For information on Freescale's Environmental Products program, go to <http://www.freescale.com/epp>.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© 2011 Freescale Semiconductor, Inc.



AUTOSAR and AUTOSAR logo are registered trademarks of AUTOSAR GbR (www.autosar.org)