

AC784xx\_DFP ADC

7.1.0

Generated by Doxygen 1.8.13

# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>File Index</b>	<b>2</b>
2.1	File List . . . . .	2
<b>3</b>	<b>Class Documentation</b>	<b>3</b>
3.1	Adc_AmoConfigType Struct Reference . . . . .	3
3.1.1	Detailed Description . . . . .	3
3.1.2	Member Data Documentation . . . . .	3
3.1.2.1	AmoInjectEn . . . . .	3
3.1.2.2	AmoInterruptEn . . . . .	4
3.1.2.3	AmoLowOffset . . . . .	4
3.1.2.4	AmoLowThreshold . . . . .	4
3.1.2.5	AmoRegularEn . . . . .	4
3.1.2.6	AmoSingleChannel . . . . .	4
3.1.2.7	AmoSingleModeEn . . . . .	5
3.1.2.8	AmoTriggerMode . . . . .	5
3.1.2.9	AmoUpOffset . . . . .	5
3.1.2.10	AmoUpThreshold . . . . .	5
3.2	Adc_Average_ConfigType Struct Reference . . . . .	5
3.2.1	Detailed Description . . . . .	6
3.2.2	Member Data Documentation . . . . .	6
3.2.2.1	Enable . . . . .	6
3.2.2.2	Value . . . . .	6

3.3	<a href="#">adc_calibration_t Struct Reference</a>	6
3.3.1	<a href="#">Detailed Description</a>	7
3.3.2	<a href="#">Member Data Documentation</a>	7
3.3.2.1	<a href="#">userGain</a>	7
3.3.2.2	<a href="#">userOffset</a>	7
3.4	<a href="#">Adc_ChainConfigType Struct Reference</a>	7
3.4.1	<a href="#">Detailed Description</a>	7
3.4.2	<a href="#">Member Data Documentation</a>	8
3.4.2.1	<a href="#">Channel</a>	8
3.4.2.2	<a href="#">InternalChannelSrc</a>	8
3.4.2.3	<a href="#">InterruptEn</a>	8
3.4.2.4	<a href="#">SeqIndex</a>	8
3.4.2.5	<a href="#">Spt</a>	8
3.5	<a href="#">Adc_ConverterConfigType Struct Reference</a>	9
3.5.1	<a href="#">Detailed Description</a>	9
3.5.2	<a href="#">Member Data Documentation</a>	9
3.5.2.1	<a href="#">Alignment</a>	9
3.5.2.2	<a href="#">ClockDivide</a>	9
3.5.2.3	<a href="#">DmaEnable</a>	10
3.5.2.4	<a href="#">HwAverage</a>	10
3.5.2.5	<a href="#">Interleave</a>	10
3.5.2.6	<a href="#">PowerEn</a>	10
3.5.2.7	<a href="#">Resolution</a>	10
3.5.2.8	<a href="#">VoltageRef</a>	11
3.6	<a href="#">Adc_GroupConfigType Struct Reference</a>	11
3.6.1	<a href="#">Detailed Description</a>	11
3.6.2	<a href="#">Member Data Documentation</a>	11
3.6.2.1	<a href="#">ContinuousModeEn</a>	12
3.6.2.2	<a href="#">DmaArgs</a>	12
3.6.2.3	<a href="#">DmaCallback</a>	12
3.6.2.4	<a href="#">DmaDstAddr</a>	12

3.6.2.5	<a href="#">InjectAutoModeEn</a>	12
3.6.2.6	<a href="#">InjectDiscontinuousModeEn</a>	13
3.6.2.7	<a href="#">InjectSequenceLength</a>	13
3.6.2.8	<a href="#">InjectTrigger</a>	13
3.6.2.9	<a href="#">IntervalModeEn</a>	13
3.6.2.10	<a href="#">RegularDiscontinuousModeEn</a>	13
3.6.2.11	<a href="#">RegularDiscontinuousNum</a>	14
3.6.2.12	<a href="#">RegularSequenceLength</a>	14
3.6.2.13	<a href="#">RegularTrigger</a>	14
3.6.2.14	<a href="#">ScanModeEn</a>	14
3.7	<a href="#">Adc_InitConfigType Struct Reference</a>	14
3.7.1	<a href="#">Detailed Description</a>	15
3.7.2	<a href="#">Member Data Documentation</a>	15
3.7.2.1	<a href="#">Interrupt</a>	15
3.8	<a href="#">Adc_InterruptInfoType Struct Reference</a>	15
3.8.1	<a href="#">Detailed Description</a>	15
3.8.2	<a href="#">Member Data Documentation</a>	15
3.8.2.1	<a href="#">Event</a>	16
3.8.2.2	<a href="#">Instance</a>	16
3.8.2.3	<a href="#">sequence</a>	16
3.9	<a href="#">Adc_InterruptType Struct Reference</a>	16
3.9.1	<a href="#">Detailed Description</a>	16
3.9.2	<a href="#">Member Data Documentation</a>	16
3.9.2.1	<a href="#">Callback</a>	16

<b>4 File Documentation</b>	<b>17</b>
4.1 AC784xx_Adc_Reg.h File Reference	17
4.1.1 Detailed Description	20
4.1.2 Macro Definition Documentation	20
4.1.2.1 ADC_EACH_REG_SPT_NUM	20
4.1.3 Function Documentation	20
4.1.3.1 Adc_Reg_BusClkEnCtrl()	20
4.1.3.2 Adc_Reg_ClearInjectEOCFlag()	21
4.1.3.3 Adc_Reg_ClearRegularEOCFlag()	21
4.1.3.4 Adc_Reg_ClearSTRFlag()	22
4.1.3.5 Adc_Reg_GetAverageEnableFlag()	22
4.1.3.6 Adc_Reg_GetAverageMode()	23
4.1.3.7 Adc_Reg_GetDMAEnableFlag()	23
4.1.3.8 Adc_Reg_GetGainOffset0Value()	24
4.1.3.9 Adc_Reg_GetGainOffset1Value()	24
4.1.3.10 Adc_Reg_GetGEOEVIN()	25
4.1.3.11 Adc_Reg_GetGEOEVINEnableFlag()	25
4.1.3.12 Adc_Reg_GetInjectData()	26
4.1.3.13 Adc_Reg_GetInjectEOCFlag()	26
4.1.3.14 Adc_Reg_GetInjectEOCInterruptEnableFlag()	27
4.1.3.15 Adc_Reg_GetInjectLength()	27
4.1.3.16 Adc_Reg_GetInjectOffset()	28
4.1.3.17 Adc_Reg_GetInjectParityVal()	28
4.1.3.18 Adc_Reg_GetRegularData()	30
4.1.3.19 Adc_Reg_GetRegularEOCFlag()	30
4.1.3.20 Adc_Reg_GetRegularEOCInterruptEnableFlag()	31
4.1.3.21 Adc_Reg_GetRegularLength()	32
4.1.3.22 Adc_Reg_GetRegularParityVal()	32
4.1.3.23 Adc_Reg_GetSTRFlag()	33
4.1.3.24 Adc_Reg_SetAMOIInjectMode()	33
4.1.3.25 Adc_Reg_SetAMOInterrupt()	34

4.1.3.26	<a href="#">Adc_Reg_SetAMOffset()</a>	34
4.1.3.27	<a href="#">Adc_Reg_SetAMORegularMode()</a>	35
4.1.3.28	<a href="#">Adc_Reg_SetAMOSingleChannel()</a>	35
4.1.3.29	<a href="#">Adc_Reg_SetAMOSingleChannelMode()</a>	36
4.1.3.30	<a href="#">Adc_Reg_SetAMOThreshold()</a>	36
4.1.3.31	<a href="#">Adc_Reg_SetAMOTriggerMode()</a>	37
4.1.3.32	<a href="#">Adc_Reg_SetAverageEnableFlag()</a>	37
4.1.3.33	<a href="#">Adc_Reg_SetAverageMode()</a>	38
4.1.3.34	<a href="#">Adc_Reg_SetCalibrationEnableFlag()</a>	38
4.1.3.35	<a href="#">Adc_Reg_SetChannelSampleTime()</a>	39
4.1.3.36	<a href="#">Adc_Reg_SetClockPrescaler()</a>	40
4.1.3.37	<a href="#">Adc_Reg_SetContinuousEnableFlag()</a>	41
4.1.3.38	<a href="#">Adc_Reg_SetDataAlign()</a>	41
4.1.3.39	<a href="#">Adc_Reg_SetDMAEnableFlag()</a>	42
4.1.3.40	<a href="#">Adc_Reg_SetGainOffset0Value()</a>	42
4.1.3.41	<a href="#">Adc_Reg_SetGainOffset1Value()</a>	43
4.1.3.42	<a href="#">Adc_Reg_SetGEOEVIN()</a>	43
4.1.3.43	<a href="#">Adc_Reg_SetGEOEVINEnableFlag()</a>	44
4.1.3.44	<a href="#">Adc_Reg_SetInjectAutoEnableFlag()</a>	44
4.1.3.45	<a href="#">Adc_Reg_SetInjectConversionChannel()</a>	45
4.1.3.46	<a href="#">Adc_Reg_SetInjectDiscontinuousEnableFlag()</a>	45
4.1.3.47	<a href="#">Adc_Reg_SetInjectEOCInterruptEnableFlag()</a>	46
4.1.3.48	<a href="#">Adc_Reg_SetInjectLength()</a>	46
4.1.3.49	<a href="#">Adc_Reg_SetInjectOffset()</a>	47
4.1.3.50	<a href="#">Adc_Reg_SetInjectTriggerSource()</a>	47
4.1.3.51	<a href="#">Adc_Reg_SetIntervalEnableFlag()</a>	48
4.1.3.52	<a href="#">Adc_Reg_SetPowerEnableFlag()</a>	49
4.1.3.53	<a href="#">Adc_Reg_SetRegularConversionChannel()</a>	49
4.1.3.54	<a href="#">Adc_Reg_SetRegularDiscontinuousEnableFlag()</a>	50
4.1.3.55	<a href="#">Adc_Reg_SetRegularDiscontinuousNum()</a>	50
4.1.3.56	<a href="#">Adc_Reg_SetRegularEOCInterruptEnableFlag()</a>	51

4.1.3.57	<a href="#">Adc_Reg_SetRegularLength()</a> . . . . .	51
4.1.3.58	<a href="#">Adc_Reg_SetRegularTriggerSource()</a> . . . . .	52
4.1.3.59	<a href="#">Adc_Reg_SetResolution()</a> . . . . .	53
4.1.3.60	<a href="#">Adc_Reg_SetScanEnableFlag()</a> . . . . .	53
4.1.3.61	<a href="#">Adc_Reg_SetVoltageReference()</a> . . . . .	54
4.1.3.62	<a href="#">Adc_Reg_SoftwareStartInjectConvert()</a> . . . . .	55
4.1.3.63	<a href="#">Adc_Reg_SoftwareStartRegularConvert()</a> . . . . .	55
4.1.3.64	<a href="#">Adc_Reg_SwRstClkCtrl()</a> . . . . .	56
4.2	<a href="#">AC784xx_API_Reference_Manual_ADC.pdf File Reference</a> . . . . .	56
4.3	<a href="#">Adc_Hal.c File Reference</a> . . . . .	56
4.3.1	<a href="#">Detailed Description</a> . . . . .	58
4.3.2	<a href="#">Macro Definition Documentation</a> . . . . .	58
4.3.2.1	<a href="#">ROUND</a> . . . . .	58
4.3.3	<a href="#">Enumeration Type Documentation</a> . . . . .	58
4.3.3.1	<a href="#">Ana_GeoecalVinType</a> . . . . .	58
4.3.4	<a href="#">Function Documentation</a> . . . . .	59
4.3.4.1	<a href="#">Adc_Hal_ClearConvertCompleteFlag()</a> . . . . .	59
4.3.4.2	<a href="#">Adc_Hal_ClearTriggerConflictFlag()</a> . . . . .	59
4.3.4.3	<a href="#">Adc_Hal_ConfigAmo()</a> . . . . .	60
4.3.4.4	<a href="#">Adc_Hal_ConfigAutoCalibration()</a> . . . . .	60
4.3.4.5	<a href="#">Adc_Hal_ConfigChannel()</a> . . . . .	61
4.3.4.6	<a href="#">Adc_Hal_ConfigConverter()</a> . . . . .	61
4.3.4.7	<a href="#">Adc_Hal_ConfigGroup()</a> . . . . .	62
4.3.4.8	<a href="#">Adc_Hal_ConfigInjectGroup()</a> . . . . .	62
4.3.4.9	<a href="#">Adc_Hal_ConfigRegularGroup()</a> . . . . .	63
4.3.4.10	<a href="#">Adc_Hal_ConvertToTemperature()</a> . . . . .	63
4.3.4.11	<a href="#">Adc_Hal_Deinit()</a> . . . . .	64
4.3.4.12	<a href="#">Adc_Hal_DmaEnable()</a> . . . . .	64
4.3.4.13	<a href="#">Adc_Hal_GetBase()</a> . . . . .	65
4.3.4.14	<a href="#">Adc_Hal_GetConvertCompleteFlag()</a> . . . . .	65
4.3.4.15	<a href="#">Adc_Hal_GetIdleFlag()</a> . . . . .	66

4.3.4.16	<a href="#">Adc_Hal_GetParityVal()</a>	66
4.3.4.17	<a href="#">Adc_Hal_GetSeqResult()</a>	67
4.3.4.18	<a href="#">Adc_Hal_GetTriggerConflictFlag()</a>	67
4.3.4.19	<a href="#">Adc_Hal_Init()</a>	68
4.3.4.20	<a href="#">Adc_Hal_InitAMOStruct()</a>	68
4.3.4.21	<a href="#">Adc_Hal_InitChanStruct()</a>	69
4.3.4.22	<a href="#">Adc_Hal_InitConverterStruct()</a>	69
4.3.4.23	<a href="#">Adc_Hal_InitGroupStruct()</a>	70
4.3.4.24	<a href="#">Adc_Hal_SetInjectOffset()</a>	70
4.3.4.25	<a href="#">Adc_Hal_SwTriggerInjectConvert()</a>	71
4.3.4.26	<a href="#">Adc_Hal_SwTriggerRegularConvert()</a>	71
4.3.4.27	<a href="#">ISR()</a> [1/2]	72
4.3.4.28	<a href="#">ISR()</a> [2/2]	72
4.4	<a href="#">Adc_Hal.h File Reference</a>	72
4.4.1	<a href="#">Detailed Description</a>	73
4.4.2	<a href="#">Function Documentation</a>	73
4.4.2.1	<a href="#">Adc_Hal_ClearConvertCompleteFlag()</a>	74
4.4.2.2	<a href="#">Adc_Hal_ClearTriggerConflictFlag()</a>	74
4.4.2.3	<a href="#">Adc_Hal_ConfigAmo()</a>	75
4.4.2.4	<a href="#">Adc_Hal_ConfigAutoCalibration()</a>	75
4.4.2.5	<a href="#">Adc_Hal_ConfigChannel()</a>	76
4.4.2.6	<a href="#">Adc_Hal_ConfigConverter()</a>	76
4.4.2.7	<a href="#">Adc_Hal_ConfigGroup()</a>	77
4.4.2.8	<a href="#">Adc_Hal_ConfigInjectGroup()</a>	77
4.4.2.9	<a href="#">Adc_Hal_ConfigRegularGroup()</a>	78
4.4.2.10	<a href="#">Adc_Hal_ConvertToTemperature()</a>	78
4.4.2.11	<a href="#">Adc_Hal_Deinit()</a>	79
4.4.2.12	<a href="#">Adc_Hal_DmaEnable()</a>	79
4.4.2.13	<a href="#">Adc_Hal_GetBase()</a>	80
4.4.2.14	<a href="#">Adc_Hal_GetConvertCompleteFlag()</a>	80
4.4.2.15	<a href="#">Adc_Hal_GetIdleFlag()</a>	81



4.4.2.16	<a href="#">Adc_Hal_GetParityVal()</a>	81
4.4.2.17	<a href="#">Adc_Hal_GetSeqResult()</a>	82
4.4.2.18	<a href="#">Adc_Hal_GetTriggerConflictFlag()</a>	82
4.4.2.19	<a href="#">Adc_Hal_Init()</a>	83
4.4.2.20	<a href="#">Adc_Hal_InitAMOStruct()</a>	83
4.4.2.21	<a href="#">Adc_Hal_InitChanStruct()</a>	84
4.4.2.22	<a href="#">Adc_Hal_InitConverterStruct()</a>	84
4.4.2.23	<a href="#">Adc_Hal_InitGroupStruct()</a>	84
4.4.2.24	<a href="#">Adc_Hal_SetInjectOffset()</a>	85
4.4.2.25	<a href="#">Adc_Hal_SwTriggerInjectConvert()</a>	85
4.4.2.26	<a href="#">Adc_Hal_SwTriggerRegularConvert()</a>	86
4.5	<a href="#">Adc_Hal_Types.h File Reference</a>	86
4.5.1	<a href="#">Detailed Description</a>	88
4.5.2	<a href="#">Macro Definition Documentation</a>	88
4.5.2.1	<a href="#">ADC_CTU_INJECT_SOURCE_COUNT</a>	88
4.5.2.2	<a href="#">ADC_CTU_REGULAR_SOURCE_COUNT</a>	88
4.5.2.3	<a href="#">ADC_GET_AVERAGE_NUM</a>	89
4.5.2.4	<a href="#">ADC_INJECT_SEQUENCE_LENGTH</a>	89
4.5.2.5	<a href="#">ADC_REGULAR_SEQUENCE_LENGTH</a>	89
4.5.2.6	<a href="#">BIT</a>	89
4.5.3	<a href="#">Typedef Documentation</a>	89
4.5.3.1	<a href="#">Adc_CallbackType</a>	89
4.5.4	<a href="#">Enumeration Type Documentation</a>	89
4.5.4.1	<a href="#">Adc_AmoTriggerModeType</a>	89
4.5.4.2	<a href="#">Adc_AverageType</a>	90
4.5.4.3	<a href="#">Adc_EventType</a>	90
4.5.4.4	<a href="#">Adc_GroupConvType</a>	90
4.5.4.5	<a href="#">Adc_InputChannelType</a>	91
4.5.4.6	<a href="#">Adc_InterleaveType</a>	92
4.5.4.7	<a href="#">Adc_PrescaleType</a>	92
4.5.4.8	<a href="#">Adc_ResolutionType</a>	93
4.5.4.9	<a href="#">Adc_ResultAlignmentType</a>	93
4.5.4.10	<a href="#">Adc_SamplingTimeType</a>	93
4.5.4.11	<a href="#">Adc_SequenceType</a>	94
4.5.4.12	<a href="#">Adc_TriggerSourceType</a>	95
4.5.4.13	<a href="#">Adc_VoltageReferenceType</a>	95

# Chapter 1

## Class Index

### 1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">Adc_AmoConfigType</a>	ADC analog monitor(AMO) config structure . . . . .	3
<a href="#">Adc_Average_ConfigType</a>	ADC hardware average structure . . . . .	5
<a href="#">adc_calibration_t</a>	ADC calibration structure . . . . .	6
<a href="#">Adc_ChanConfigType</a>	Defines the control Channel configuration . . . . .	7
<a href="#">Adc_ConverterConfigType</a>	ADC geneneral conversion config . . . . .	9
<a href="#">Adc_GroupConfigType</a>	Defines the control group configuration . . . . .	11
<a href="#">Adc_InitConfigType</a>	ADC init config args structure . . . . .	14
<a href="#">Adc_InterruptInfoType</a>	ADC interrupt information structure . . . . .	15
<a href="#">Adc_InterruptType</a>	ADC interrupt struct . . . . .	16

# Chapter 2

## File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<a href="#">AC784xx_Adc_Reg.h</a>	
Adc access register inline function definition . . . . .	17
<a href="#">AC784xx_API_Reference_Manual_ADC.pdf</a> . . . . .	56
<a href="#">Adc_Hal.c</a>	
This file provides all adc hal api . . . . .	56
<a href="#">Adc_Hal.h</a>	
This file provides all adc hal api . . . . .	72
<a href="#">Adc_Hal_Types.h</a>	
This file provides adc hal types header . . . . .	86

## Chapter 3

# Class Documentation

### 3.1 Adc\_AmoConfigType Struct Reference

ADC analog monitor(AMO) config structure.

```
#include <Adc_Hal_Types.h>
```

#### Public Attributes

- [Adc\\_AmoTriggerModeType](#) AmoTriggerMode
- boolean [AmoInterruptEn](#)
- boolean [AmoRegularEn](#)
- boolean [AmoInjectEn](#)
- boolean [AmoSingleModeEn](#)
- [Adc\\_InputChannelType](#) AmoSingleChannel
- uint16 [AmoUpThreshold](#)
- uint16 [AmoLowThreshold](#)
- uint16 [AmoUpOffset](#)
- uint16 [AmoLowOffset](#)

#### 3.1.1 Detailed Description

ADC analog monitor(AMO) config structure.

Definition at line 445 of file Adc\_Hal\_Types.h.

#### 3.1.2 Member Data Documentation

##### 3.1.2.1 AmoInjectEn

```
boolean Adc_AmoConfigType::AmoInjectEn
```

Enable/disable analog monitor injected group mode

Definition at line 450 of file Adc\_Hal\_Types.h.

### 3.1.2.2 AmoInterruptEn

```
boolean Adc_AmoConfigType::AmoInterruptEn
```

Enable/disable AMO interrupt

Definition at line 448 of file Adc\_Hal\_Types.h.

### 3.1.2.3 AmoLowOffset

```
uint16 Adc_AmoConfigType::AmoLowOffset
```

Set analog monitor lower Offset

Definition at line 456 of file Adc\_Hal\_Types.h.

### 3.1.2.4 AmoLowThreshold

```
uint16 Adc_AmoConfigType::AmoLowThreshold
```

Set analog monitor lower Threshold

Definition at line 454 of file Adc\_Hal\_Types.h.

### 3.1.2.5 AmoRegularEn

```
boolean Adc_AmoConfigType::AmoRegularEn
```

Enable/disable analog monitor regular group mode

Definition at line 449 of file Adc\_Hal\_Types.h.

### 3.1.2.6 AmoSingleChannel

```
Adc\_InputChannelType Adc_AmoConfigType::AmoSingleChannel
```

Select analog monitor single Channel

Definition at line 452 of file Adc\_Hal\_Types.h.

### 3.1.2.7 AmoSingleModeEn

```
boolean Adc_AmoConfigType::AmoSingleModeEn
```

Enable/disable analog monitor single Channel mode

Definition at line 451 of file Adc\_Hal\_Types.h.

### 3.1.2.8 AmoTriggerMode

```
Adc_AmoTriggerModeType Adc_AmoConfigType::AmoTriggerMode
```

0:level trigger; 1:edge trigger

Definition at line 447 of file Adc\_Hal\_Types.h.

### 3.1.2.9 AmoUpOffset

```
uint16 Adc_AmoConfigType::AmoUpOffset
```

Set analog monitor upper Offset

Definition at line 455 of file Adc\_Hal\_Types.h.

### 3.1.2.10 AmoUpThreshold

```
uint16 Adc_AmoConfigType::AmoUpThreshold
```

Set analog monitor upper Threshold

Definition at line 453 of file Adc\_Hal\_Types.h.

The documentation for this struct was generated from the following file:

- [Adc\\_Hal\\_Types.h](#)

## 3.2 Adc\_Average\_ConfigType Struct Reference

ADC hardware average structure.

```
#include <Adc_Hal_Types.h>
```

## Public Attributes

- boolean [Enable](#)
- [Adc\\_AverageType](#) Value

### 3.2.1 Detailed Description

ADC hardware average structure.

Definition at line 373 of file `Adc_Hal_Types.h`.

### 3.2.2 Member Data Documentation

#### 3.2.2.1 Enable

```
boolean Adc_Average_ConfigType::Enable
```

Enable hardware averaging

Definition at line 375 of file `Adc_Hal_Types.h`.

#### 3.2.2.2 Value

```
Adc\_AverageType Adc_Average_ConfigType::Value
```

Selection for number of samples used for hardware averaging

Definition at line 376 of file `Adc_Hal_Types.h`.

The documentation for this struct was generated from the following file:

- [Adc\\_Hal\\_Types.h](#)

## 3.3 adc\_calibration\_t Struct Reference

ADC calibration structure.

## Public Attributes

- sint16 [userGain](#)
- sint16 [userOffset](#)

### 3.3.1 Detailed Description

ADC calibration structure.

Definition at line 67 of file Adc\_Hal.c.

### 3.3.2 Member Data Documentation

#### 3.3.2.1 userGain

```
sint16 adc_calibration_t::userGain
```

Definition at line 69 of file Adc\_Hal.c.

#### 3.3.2.2 userOffset

```
sint16 adc_calibration_t::userOffset
```

Definition at line 70 of file Adc\_Hal.c.

The documentation for this struct was generated from the following file:

- [Adc\\_Hal.c](#)

## 3.4 Adc\_ChanConfigType Struct Reference

Defines the control Channel configuration.

```
#include <Adc_Hal_Types.h>
```

### Public Attributes

- [Adc\\_InputChannelType](#) Channel
- [Adc\\_SamplingTimeType](#) Spt
- [Adc\\_SequenceType](#) SeqIndex
- boolean [InterruptEn](#)
- [Adc\\_InternalChannelSrcType](#) [InternalChannelSrc](#)

### 3.4.1 Detailed Description

Defines the control Channel configuration.

Definition at line 412 of file Adc\_Hal\_Types.h.



### 3.4.2 Member Data Documentation

#### 3.4.2.1 Channel

[Adc\\_InputChannelType](#) Adc\_ChanConfigType::Channel

Selection of input Channel for measurement

Definition at line 414 of file Adc\_Hal\_Types.h.

#### 3.4.2.2 InternalChannelSrc

[Adc\\_InternalChannelSrcType](#) Adc\_ChanConfigType::InternalChannelSrc

slection of internal channel srouce

Definition at line 418 of file Adc\_Hal\_Types.h.

#### 3.4.2.3 InterruptEn

[boolean](#) Adc\_ChanConfigType::InterruptEn

Enable/disbale EOC irq for the sequeunce index

Definition at line 417 of file Adc\_Hal\_Types.h.

#### 3.4.2.4 SeqIndex

[Adc\\_SequenceType](#) Adc\_ChanConfigType::SeqIndex

Group sequeunce index

Definition at line 416 of file Adc\_Hal\_Types.h.

#### 3.4.2.5 Spt

[Adc\\_SamplingTimeType](#) Adc\_ChanConfigType::Spt

Sample time

Definition at line 415 of file Adc\_Hal\_Types.h.

The documentation for this struct was generated from the following file:

- [Adc\\_Hal\\_Types.h](#)

## 3.5 Adc\_ConverterConfigType Struct Reference

ADC general conversion config.

```
#include <Adc_Hal_Types.h>
```

### Public Attributes

- [Adc\\_Average\\_ConfigType](#) HwAverage
- [Adc\\_InterleaveType](#) Interleave
- [Adc\\_PrescaleType](#) ClockDivide
- [Adc\\_ResolutionType](#) Resolution
- [Adc\\_ResultAlignmentType](#) Alignment
- [Adc\\_VoltageReferenceType](#) VoltageRef
- boolean [DmaEnable](#)
- boolean [PowerEn](#)

### 3.5.1 Detailed Description

ADC general conversion config.

Definition at line 397 of file `Adc_Hal_Types.h`.

### 3.5.2 Member Data Documentation

#### 3.5.2.1 Alignment

[Adc\\_ResultAlignmentType](#) `Adc_ConverterConfigType::Alignment`

ADC result Alignment setting

Definition at line 403 of file `Adc_Hal_Types.h`.

#### 3.5.2.2 ClockDivide

[Adc\\_PrescaleType](#) `Adc_ConverterConfigType::ClockDivide`

ADC clock divide

Definition at line 401 of file `Adc_Hal_Types.h`.

### 3.5.2.3 DmaEnable

```
boolean Adc_ConverterConfigType::DmaEnable
```

Enable/disable ADC DMA support function

Definition at line 405 of file Adc\_Hal\_Types.h.

### 3.5.2.4 HwAverage

```
Adc_Average_ConfigType Adc_ConverterConfigType::HwAverage
```

ADC HW average setting

Definition at line 399 of file Adc\_Hal\_Types.h.

### 3.5.2.5 Interleave

```
Adc_InterleaveType Adc_ConverterConfigType::Interleave
```

ADC Interleave setting

Definition at line 400 of file Adc\_Hal\_Types.h.

### 3.5.2.6 PowerEn

```
boolean Adc_ConverterConfigType::PowerEn
```

Enable/disable ADC analog module power Supply

Definition at line 406 of file Adc\_Hal\_Types.h.

### 3.5.2.7 Resolution

```
Adc_ResolutionType Adc_ConverterConfigType::Resolution
```

ADC Resolution setting

Definition at line 402 of file Adc\_Hal\_Types.h.

### 3.5.2.8 VoltageRef

[Adc\\_VoltageReferenceType](#) `Adc_ConverterConfigType::VoltageRef`

ADC reference source selection

Definition at line 404 of file `Adc_Hal_Types.h`.

The documentation for this struct was generated from the following file:

- [Adc\\_Hal\\_Types.h](#)

## 3.6 Adc\_GroupConfigType Struct Reference

Defines the control group configuration.

```
#include <Adc_Hal_Types.h>
```

### Public Attributes

- `uint32` [DmaDstAddr](#)
- [Adc\\_TriggerSourceType](#) `RegularTrigger`
- [Adc\\_TriggerSourceType](#) `InjectTrigger`
- `uint8` [RegularSequenceLength](#)
- `uint8` [InjectSequenceLength](#)
- `uint8` [RegularDiscontinuousNum](#)
- `boolean` [ScanModeEn](#)
- `boolean` [ContinuousModeEn](#)
- `boolean` [RegularDiscontinuousModeEn](#)
- `boolean` [InjectDiscontinuousModeEn](#)
- `boolean` [InjectAutoModeEn](#)
- `boolean` [IntervalModeEn](#)
- `Hal_CallbackType` [DmaCallback](#)
- `void *` [DmaArgs](#)

### 3.6.1 Detailed Description

Defines the control group configuration.

Definition at line 424 of file `Adc_Hal_Types.h`.

### 3.6.2 Member Data Documentation

### 3.6.2.1 ContinuousModeEn

```
boolean Adc_GroupConfigType::ContinuousModeEn
```

Enable/disable continuous mode

Definition at line 433 of file Adc\_Hal\_Types.h.

### 3.6.2.2 DmaArgs

```
void* Adc_GroupConfigType::DmaArgs
```

Dma callback args

Definition at line 439 of file Adc\_Hal\_Types.h.

### 3.6.2.3 DmaCallback

```
Hal_CallbackType Adc_GroupConfigType::DmaCallback
```

Dma callback

Definition at line 438 of file Adc\_Hal\_Types.h.

### 3.6.2.4 DmaDstAddr

```
uint32 Adc_GroupConfigType::DmaDstAddr
```

Definition at line 426 of file Adc\_Hal\_Types.h.

### 3.6.2.5 InjectAutoModeEn

```
boolean Adc_GroupConfigType::InjectAutoModeEn
```

Enable/disable automatic mode for injected group

Definition at line 436 of file Adc\_Hal\_Types.h.

### 3.6.2.6 InjectDiscontinuousModeEn

```
boolean Adc_GroupConfigType::InjectDiscontinuousModeEn
```

Enable/disable discontinuous mode for injected group

Definition at line 435 of file Adc\_Hal\_Types.h.

### 3.6.2.7 InjectSequenceLength

```
uint8 Adc_GroupConfigType::InjectSequenceLength
```

Injected sequence length

Definition at line 430 of file Adc\_Hal\_Types.h.

### 3.6.2.8 InjectTrigger

```
Adc\_TriggerSourceType Adc_GroupConfigType::InjectTrigger
```

injection group trigger source type

Definition at line 428 of file Adc\_Hal\_Types.h.

### 3.6.2.9 IntervalModeEn

```
boolean Adc_GroupConfigType::IntervalModeEn
```

Enable/disable interval mode for mode3/5

Definition at line 437 of file Adc\_Hal\_Types.h.

### 3.6.2.10 RegularDiscontinuousModeEn

```
boolean Adc_GroupConfigType::RegularDiscontinuousModeEn
```

Enable/disable discontinuous mode for regular group

Definition at line 434 of file Adc\_Hal\_Types.h.

#### 3.6.2.11 RegularDiscontinuousNum

```
uint8 Adc_GroupConfigType::RegularDiscontinuousNum
```

Regular discontinuous mode number

Definition at line 431 of file Adc\_Hal\_Types.h.

#### 3.6.2.12 RegularSequenceLength

```
uint8 Adc_GroupConfigType::RegularSequenceLength
```

Regular sequence length

Definition at line 429 of file Adc\_Hal\_Types.h.

#### 3.6.2.13 RegularTrigger

```
Adc_TriggerSourceType Adc_GroupConfigType::RegularTrigger
```

regular group trigger source type

Definition at line 427 of file Adc\_Hal\_Types.h.

#### 3.6.2.14 ScanModeEn

```
boolean Adc_GroupConfigType::ScanModeEn
```

Enable/disable scan mode for regular and injected group

Definition at line 432 of file Adc\_Hal\_Types.h.

The documentation for this struct was generated from the following file:

- [Adc\\_Hal\\_Types.h](#)

## 3.7 Adc\_InitConfigType Struct Reference

ADC init config args structure.

```
#include <Adc_Hal_Types.h>
```

### Public Attributes

- [Adc\\_InterruptType Interrupt](#)

### 3.7.1 Detailed Description

ADC init config args structure.

Definition at line 470 of file Adc\_Hal\_Types.h.

### 3.7.2 Member Data Documentation

#### 3.7.2.1 Interrupt

[Adc\\_InterruptType](#) Adc\_InitConfigType::Interrupt

ADC irq args

Definition at line 472 of file Adc\_Hal\_Types.h.

The documentation for this struct was generated from the following file:

- [Adc\\_Hal\\_Types.h](#)

## 3.8 Adc\_InterruptInfoType Struct Reference

ADC interrupt information structure.

```
#include <Adc_Hal_Types.h>
```

### Public Attributes

- uint8 [Instance](#)
- uint32 [Event](#)
- [Adc\\_SequenceType](#) sequence

### 3.8.1 Detailed Description

ADC interrupt information structure.

Definition at line 382 of file Adc\_Hal\_Types.h.

### 3.8.2 Member Data Documentation



### 3.8.2.1 Event

`uint32 Adc_InterruptInfoType::Event`

adc Event bit mask, combination of `Adc_EventType`

Definition at line 385 of file `Adc_Hal_Types.h`.

### 3.8.2.2 Instance

`uint8 Adc_InterruptInfoType::Instance`

ADC Instance number

Definition at line 384 of file `Adc_Hal_Types.h`.

### 3.8.2.3 sequence

`Adc_SequenceType Adc_InterruptInfoType::sequence`

Group sequence index

Definition at line 386 of file `Adc_Hal_Types.h`.

The documentation for this struct was generated from the following file:

- [Adc\\_Hal\\_Types.h](#)

## 3.9 Adc\_InterruptType Struct Reference

ADC interrupt struct.

```
#include <Adc_Hal_Types.h>
```

### Public Attributes

- [Adc\\_CallbackType Callback](#)

### 3.9.1 Detailed Description

ADC interrupt struct.

Definition at line 462 of file `Adc_Hal_Types.h`.

### 3.9.2 Member Data Documentation

#### 3.9.2.1 Callback

`Adc_CallbackType Adc_InterruptType::Callback`

ADC irq callback

Definition at line 464 of file `Adc_Hal_Types.h`.

The documentation for this struct was generated from the following file:

- [Adc\\_Hal\\_Types.h](#)

## Chapter 4

# File Documentation

### 4.1 AC784xx\_Adc\_Reg.h File Reference

Adc access register inline function definition.

```
#include "Adc_Hal_Types.h"
#include "Device_Register.h"
```

#### Macros

- #define [ADC\\_EACH\\_REG\\_SPT\\_NUM](#) (10UL)

#### Functions

- LOCAL\_INLINE void [Adc\\_Reg\\_SetClockPrescaler](#) (ADC\_Type \*const Base, const [Adc\\_PrescaleType](#) psc)  
*Set adc clock prescaler.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetResolution](#) (ADC\_Type \*const Base, const [Adc\\_ResolutionType](#) Resolution)  
*Set adc Resolution.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetVoltageReference](#) (ADC\_Type \*const Base, const [Adc\\_VoltageReferenceType](#) VoltageRef)  
*Set adc voltage reference source.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetPowerEnableFlag](#) (ADC\_Type \*const Base, const boolean mode)  
*Set adc power mode.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetScanEnableFlag](#) (ADC\_Type \*const Base, const boolean State)  
*Set scan convert mode enable flag.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetIntervalEnableFlag](#) (ADC\_Type \*const Base, const boolean State)  
*Set ADC interval mode.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetContinuousEnableFlag](#) (ADC\_Type \*const Base, const boolean State)  
*Set continuous convert mode.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetRegularDiscontinuousEnableFlag](#) (ADC\_Type \*const Base, const boolean State)  
*Set discontinuous mode enable flag of regular group.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetRegularDiscontinuousNum](#) (ADC\_Type \*const Base, const uint8 num)  
*Set discontinuous mode number for regular group.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetInjectAutoEnableFlag](#) (ADC\_Type \*const Base, const boolean State)  
*Set ADC automatic inject injection group convert after regular group function enable flag.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetInjectDiscontinuousEnableFlag](#) (ADC\_Type \*const Base, const boolean State)

- Set discontinuous mode for injection group.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetRegularTriggerSource](#) (ADC\_Type \*const Base, const [Adc\\_TriggerSourceType](#) source)
- Set regular group trigger source.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SoftwareStartRegularConvert](#) (ADC\_Type \*const Base)
- Start software trigger for regular group convert.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SoftwareStartInjectConvert](#) (ADC\_Type \*const Base)
- Start software trigger for inject group convert.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetInjectTriggerSource](#) (ADC\_Type \*const Base, const [Adc\\_TriggerSourceType](#) source)
- Set injected group trigger source.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetInjectOffset](#) (ADC\_Type \*const Base, uint8 Num, uint16 Offset)
- Set injection group offset Value.*
- LOCAL\_INLINE uint16 [Adc\\_Reg\\_GetInjectOffset](#) (const ADC\_Type \*const Base, const uint8 Num)
- Get injection group offset value.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetRegularLength](#) (ADC\_Type \*const Base, const uint8 length)
- Set regular group conversion length.*
- LOCAL\_INLINE uint8 [Adc\\_Reg\\_GetRegularLength](#) (ADC\_Type \*const Base)
- Get regular group conversion length.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetInjectLength](#) (ADC\_Type \*const Base, const uint8 length)
- Set injection group conversion length.*
- LOCAL\_INLINE uint8 [Adc\\_Reg\\_GetInjectLength](#) (ADC\_Type \*const Base)
- Get injection group conversion length.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetChannelSampleTime](#) (ADC\_Type \*const Base, const [Adc\\_InputChannelType](#) Channel, const [Adc\\_SamplingTimeType](#) Time)
- Set Channel sample time.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetDMAEnableFlag](#) (ADC\_Type \*const Base, boolean State)
- Set ADC DMA enable flag.*
- LOCAL\_INLINE boolean [Adc\\_Reg\\_GetDMAEnableFlag](#) (ADC\_Type \*const Base)
- Get ADC DMA enable flag.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetRegularConversionChannel](#) (ADC\_Type \*const Base, const [Adc\\_SequenceType](#) Seq, [Adc\\_InputChannelType](#) Channel)
- Set regular group sequence.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetRegularEOCInterruptEnableFlag](#) (ADC\_Type \*const Base, const [Adc\\_SequenceType](#) Seq, const boolean State)
- Set regular group end of conversion(EOC) interrupt enable flag.*
- LOCAL\_INLINE boolean [Adc\\_Reg\\_GetRegularEOCInterruptEnableFlag](#) (ADC\_Type \*const Base, const [Adc\\_SequenceType](#) Seq)
- Get regular group EOC interrupt enable flag.*
- LOCAL\_INLINE boolean [Adc\\_Reg\\_GetRegularEOCFlag](#) (ADC\_Type \*const Base, const [Adc\\_SequenceType](#) Seq)
- Get regular group EOC flag.*
- LOCAL\_INLINE void [Adc\\_Reg\\_ClearRegularEOCFlag](#) (ADC\_Type \*const Base, const [Adc\\_SequenceType](#) Seq)
- Clear regular group EOC flag.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetInjectConversionChannel](#) (ADC\_Type \*const Base, const [Adc\\_SequenceType](#) Seq, [Adc\\_InputChannelType](#) Channel)
- Set injection group sequence.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetInjectEOCInterruptEnableFlag](#) (ADC\_Type \*const Base, const [Adc\\_SequenceType](#) Seq, const boolean State)
- Set injection group end of conversion(EOC) interrupt enable flag.*
- LOCAL\_INLINE boolean [Adc\\_Reg\\_GetInjectEOCInterruptEnableFlag](#) (ADC\_Type \*const Base, const [Adc\\_SequenceType](#) Seq)
- Get injection group EOC interrupt enable flag.*
- LOCAL\_INLINE boolean [Adc\\_Reg\\_GetInjectEOCFlag](#) (ADC\_Type \*const Base, const [Adc\\_SequenceType](#) Seq)
- Get injection group EOC flag.*

- LOCAL\_INLINE void [Adc\\_Reg\\_ClearInjectEOCFlag](#) (ADC\_Type \*const Base, const [Adc\\_SequenceType](#) Seq)  
*Clear injection group EOC flag.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetDataAlign](#) (ADC\_Type \*const Base, const [Adc\\_ResultAlignmentType](#) align)  
*Set ADC data Alignment.*
- LOCAL\_INLINE uint16 [Adc\\_Reg\\_GetRegularData](#) (ADC\_Type \*const Base, const [Adc\\_SequenceType](#) Seq)  
*Get ADC regular group data.*
- LOCAL\_INLINE uint8 [Adc\\_Reg\\_GetRegularParityVal](#) (ADC\_Type \*const Base, const [Adc\\_SequenceType](#) Seq)  
*Get ADC regular group parity val.*
- LOCAL\_INLINE uint16 [Adc\\_Reg\\_GetInjectData](#) (ADC\_Type \*const Base, const [Adc\\_SequenceType](#) Seq)  
*Get ADC injection data.*
- LOCAL\_INLINE uint8 [Adc\\_Reg\\_GetInjectParityVal](#) (ADC\_Type \*const Base, const [Adc\\_SequenceType](#) Seq)  
*Get ADC injection group parity val.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetAverageEnableFlag](#) (ADC\_Type \*const Base, boolean State)  
*Set hardware average enable flag.*
- LOCAL\_INLINE boolean [Adc\\_Reg\\_GetAverageEnableFlag](#) (const ADC\_Type \*const Base)  
*Get hardware average enable flag.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetAverageMode](#) (ADC\_Type \*const Base, uint8 averageMode)  
*Set hardware average mode.*
- LOCAL\_INLINE uint8 [Adc\\_Reg\\_GetAverageMode](#) (const ADC\_Type \*const Base)  
*Get hardware average mode.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SwRstClkCtrl](#) (uint32 Instance, boolean IsEnable)  
*Adc\_Reg\_SwRstClkCtrl.*
- LOCAL\_INLINE void [Adc\\_Reg\\_BusClkEnCtrl](#) (uint32 Instance, boolean IsEnable)  
*BusClk Enable Ctrl.*
- LOCAL\_INLINE uint32 [Adc\\_Reg\\_GetSTRFlag](#) (ADC\_Type \*const Base)  
*Get STR reg flag.*
- LOCAL\_INLINE void [Adc\\_Reg\\_ClearSTRFlag](#) (ADC\_Type \*const Base, uint32 Mask)  
*Clear STR reg flag.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetAMOIInterrupt](#) (ADC\_Type \*const Base, const boolean State)  
*Set AMO Interrupt.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetAMOSingleChannel](#) (ADC\_Type \*const Base, const [Adc\\_InputChannelType](#) Channel)  
*Set analog monitor single channel.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetAMOSingleChannelMode](#) (ADC\_Type \*const Base, const boolean State)  
*Set analog monitor single channel mode.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetAMORegularMode](#) (ADC\_Type \*const Base, const boolean State)  
*Set regular group analog monitor mode.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetAMOInjectMode](#) (ADC\_Type \*const Base, const boolean State)  
*Set injected group analog monitor mode.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetAMOTriggerMode](#) (ADC\_Type \*const Base, [Adc\\_AmoTriggerModeType](#) Mode)  
*Set analog monitor trigger type.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetAMOThreshold](#) (ADC\_Type \*const Base, uint16 HighValue, uint16 LowValue)  
*Set analog monitor threshold.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetAMOffset](#) (ADC\_Type \*const Base, uint16 HighOffset, uint16 LowOffset)  
*Set analog monitor offset.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetCalibrationEnableFlag](#) (ADC\_Type \*const Base, boolean State)  
*Set Calibration function enable flag.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetGEOEVINEnableFlag](#) (uint8 Instance, boolean State)  
*Set GE OE calibration voltage input signal enable flag.*
- LOCAL\_INLINE boolean [Adc\\_Reg\\_GetGEOEVINEnableFlag](#) (uint8 Instance)  
*Get GE OE calibration voltage input signal enable flag.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetGEOEVIN](#) (uint8 Instance, uint8 Vin)  
*Set GE OE calibration input signal.*

- LOCAL\_INLINE uint8 [Adc\\_Reg\\_GetGEOEVIN](#) (uint8 Instance)  
*Get GE OE calibration input signal.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetGainOffset0Value](#) (ADC\_Type \*const Base, sint16 Gain, sint16 Offset)  
*Set gain error and offset error value0 for external channel.*
- LOCAL\_INLINE void [Adc\\_Reg\\_GetGainOffset0Value](#) (const ADC\_Type \*const Base, sint16 \*const Gain, sint16 \*const Offset)  
*Get gain error and offset error value0 for external channel.*
- LOCAL\_INLINE void [Adc\\_Reg\\_SetGainOffset1Value](#) (ADC\_Type \*const Base, sint16 Gain, sint16 Offset)  
*Set gain error and offset error value0 for internal channel.*
- LOCAL\_INLINE void [Adc\\_Reg\\_GetGainOffset1Value](#) (const ADC\_Type \*const Base, sint16 \*const Gain, sint16 \*const Offset)  
*Get gain error and offset error value0 for internal channel.*

### 4.1.1 Detailed Description

Adc access register inline function definition.

### 4.1.2 Macro Definition Documentation

#### 4.1.2.1 ADC\_EACH\_REG\_SPT\_NUM

```
#define ADC_EACH_REG_SPT_NUM (10UL)
```

ADC sampel timer setting number in each SPT register

Definition at line 50 of file AC784xx\_Adc\_Reg.h.

### 4.1.3 Function Documentation

#### 4.1.3.1 Adc\_Reg\_BusClkEnCtrl()

```
LOCAL_INLINE void Adc_Reg_BusClkEnCtrl (
    uint32 Instance,
    boolean IsEnable )
```

BusClk Enable Ctrl.

#### Note

Function ID: DES\_ADC\_API\_607  
Service ID: none

**Parameters**

in	<i>Instance</i>	adc module instance
in	<i>IsEnable</i>	enable flag

**Returns**

none

Definition at line 863 of file AC784xx\_Adc\_Reg.h.

**4.1.3.2 Adc\_Reg\_ClearInjectEOCFlag()**

```
LOCAL_INLINE void Adc_Reg_ClearInjectEOCFlag (
    ADC_Type *const Base,
    const Adc_SequenceType Seq )
```

Clear injection group EOC flag.

**Note**

Function ID: DES\_ADC\_API\_558

Service ID: none

**Parameters**

in	<i>Base</i>	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	<i>Seq</i>	adc sequence

**Returns**

none

Definition at line 687 of file AC784xx\_Adc\_Reg.h.

**4.1.3.3 Adc\_Reg\_ClearRegularEOCFlag()**

```
LOCAL_INLINE void Adc_Reg_ClearRegularEOCFlag (
    ADC_Type *const Base,
    const Adc_SequenceType Seq )
```

Clear regular group EOC flag.

**Note**

Function ID: DES\_ADC\_API\_552

Service ID: none

**Parameters**

in	<i>Base</i>	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	<i>Seq</i>	adc sequence

**Returns**

none

Definition at line 588 of file AC784xx\_Adc\_Reg.h.

**4.1.3.4 Adc\_Reg\_ClearSTRFlag()**

```
LOCAL_INLINE void Adc_Reg_ClearSTRFlag (  
    ADC_Type *const Base,  
    uint32 Mask )
```

Clear STR reg flag.

**Note**

Function ID: DES\_ADC\_API\_610  
Service ID: none

**Parameters**

in	<i>Base</i>	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	<i>Mask</i>	Register value.

**Returns**

analog monitor flag State

Definition at line 898 of file AC784xx\_Adc\_Reg.h.

**4.1.3.5 Adc\_Reg\_GetAverageEnableFlag()**

```
LOCAL_INLINE boolean Adc_Reg_GetAverageEnableFlag (  
    const ADC_Type *const Base )
```

Get hardware average enable flag.

**Parameters**

in	Base	adc module
		<ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>

**Returns**

whether hardware average function is enabled

Definition at line 796 of file AC784xx\_Adc\_Reg.h.

**4.1.3.6 Adc\_Reg\_GetAverageMode()**

```
LOCAL_INLINE uint8 Adc_Reg_GetAverageMode (  
    const ADC_Type *const Base )
```

Get hardware average mode.

**Parameters**

in	Base	adc module
		<ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>

**Returns**

average mode

- ADC\_AVERAGE\_4
- ADC\_AVERAGE\_8
- ADC\_AVERAGE\_16
- ADC\_AVERAGE\_32

Definition at line 832 of file AC784xx\_Adc\_Reg.h.

**4.1.3.7 Adc\_Reg\_GetDMAEnableFlag()**

```
LOCAL_INLINE boolean Adc_Reg_GetDMAEnableFlag (  
    ADC_Type *const Base )
```

Get ADC DMA enable flag.

**Note**

Function ID: DES\_ADC\_API\_545  
Service ID: none



**Parameters**

in	<i>Base</i>	adc module
		<ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>

**Returns**

enable or disable DMA function in ADC

Definition at line 492 of file AC784xx\_Adc\_Reg.h.

**4.1.3.8 Adc\_Reg\_GetGainOffset0Value()**

```
LOCAL_INLINE void Adc_Reg_GetGainOffset0Value (
    const ADC_Type *const Base,
    sint16 *const Gain,
    sint16 *const Offset )
```

Get gain error and offset error value0 for external channel.

**Parameters**

in	<i>Base</i>	adc module
		<ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
out	<i>Gain</i>	gain error in LSB.
out	<i>Offset</i>	offset error in LSB.

**Returns**

none

Definition at line 1214 of file AC784xx\_Adc\_Reg.h.

**4.1.3.9 Adc\_Reg\_GetGainOffset1Value()**

```
LOCAL_INLINE void Adc_Reg_GetGainOffset1Value (
    const ADC_Type *const Base,
    sint16 *const Gain,
    sint16 *const Offset )
```

Get gain error and offset error value0 for internal channel.

**Parameters**

in	<i>Base</i>	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	<i>Gain</i>	gain error in LSB.
in	<i>Offset</i>	offset error in LSB.

**Returns**

none

Definition at line 1247 of file AC784xx\_Adc\_Reg.h.

**4.1.3.10 Adc\_Reg\_GetGEOEVIN()**

```
LOCAL_INLINE uint8 Adc_Reg_GetGEOEVIN (
    uint8 Instance )
```

Get GE OE calibration input signal.

**Parameters**

in	<i>Instance</i>	adc instance number
----	-----------------	---------------------

**Returns**

calibration input signal

Definition at line 1157 of file AC784xx\_Adc\_Reg.h.

**4.1.3.11 Adc\_Reg\_GetGEOEVINEnableFlag()**

```
LOCAL_INLINE boolean Adc_Reg_GetGEOEVINEnableFlag (
    uint8 Instance )
```

Get GE OE calibration voltage input signal enable flag.

**Parameters**

in	<i>Instance</i>	adc instance number
----	-----------------	---------------------

**Returns**

whether calibration voltage input is enabled

Definition at line 1087 of file AC784xx\_Adc\_Reg.h.

4.1.3.12 Adc\_Reg\_GetInjectData()

```
LOCAL_INLINE uint16 Adc_Reg_GetInjectData (
    ADC_Type *const Base,
    const Adc_SequenceType Seq )
```

Get ADC injection data.

Note

Function ID: DES\_ADC\_API\_564  
Service ID: none

Parameters

in	Base	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	Seq	adc sequeunce

Returns

ADC data of particular sequence

Definition at line 753 of file AC784xx\_Adc\_Reg.h.

4.1.3.13 Adc\_Reg\_GetInjectEOCFlag()

```
LOCAL_INLINE boolean Adc_Reg_GetInjectEOCFlag (
    ADC_Type *const Base,
    const Adc_SequenceType Seq )
```

Get injection group EOC flag.

Note

Function ID: DES\_ADC\_API\_557  
Service ID: none

**Parameters**

in	<i>Base</i>	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	<i>Seq</i>	adc sequence

**Returns**

EOC flag of particular sequence

Definition at line 667 of file AC784xx\_Adc\_Reg.h.

**4.1.3.14 Adc\_Reg\_GetInjectEOCInterruptEnableFlag()**

```
LOCAL_INLINE boolean Adc_Reg_GetInjectEOCInterruptEnableFlag (  
    ADC_Type *const Base,  
    const Adc_SequenceType Seq )
```

Get injection group EOC interrupt enable flag.

**Note**

Function ID: DES\_ADC\_API\_556  
Service ID: none

**Parameters**

in	<i>Base</i>	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	<i>Seq</i>	adc sequence

**Returns**

whether EOC interrupt of particular sequence is enabled

Definition at line 651 of file AC784xx\_Adc\_Reg.h.

**4.1.3.15 Adc\_Reg\_GetInjectLength()**

```
LOCAL_INLINE uint8 Adc_Reg_GetInjectLength (  
    ADC_Type *const Base )
```

Get injection group conversion length.

Note

Function ID: DES\_ADC\_API\_540  
Service ID: none

Parameters

<i>in</i>	<i>Base</i>	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
-----------	-------------	--

Returns

injection group length setting

Definition at line 409 of file AC784xx\_Adc\_Reg.h.

4.1.3.16 Adc\_Reg\_GetInjectOffset()

```
LOCAL_INLINE uint16 Adc_Reg_GetInjectOffset (
    const ADC_Type *const Base,
    const uint8 Num )
```

Get injection group offset value.

Parameters

<i>in</i>	<i>Base</i>	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
<i>in</i>	<i>Num</i>	injection group index, should be less than ADC_INJECT_SEQUENCE_LEGNTH

Returns

offset value

Definition at line 349 of file AC784xx\_Adc\_Reg.h.

4.1.3.17 Adc\_Reg\_GetInjectParityVal()

```
LOCAL_INLINE uint8 Adc_Reg_GetInjectParityVal (
    ADC_Type *const Base,
    const Adc\_SequenceType Seq )
```

Get ADC injection group parity val.

**Note**

Function ID: DES\_ADC\_API\_561  
Service ID: none

**Parameters**

in	<i>Base</i>	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	<i>Seq</i>	adc sequence

**Returns**

ADC parity val of particular sequence

Definition at line 768 of file AC784xx\_Adc\_Reg.h.

**4.1.3.18 Adc\_Reg\_GetRegularData()**

```
LOCAL_INLINE uint16 Adc_Reg_GetRegularData (
    ADC_Type *const Base,
    const Adc_SequenceType Seq )
```

Get ADC regular group data.

**Note**

Function ID: DES\_ADC\_API\_561  
Service ID: none

**Parameters**

in	<i>Base</i>	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	<i>Seq</i>	adc sequence

**Returns**

ADC data of particular sequence

Definition at line 723 of file AC784xx\_Adc\_Reg.h.

**4.1.3.19 Adc\_Reg\_GetRegularEOCFlag()**

```
LOCAL_INLINE boolean Adc_Reg_GetRegularEOCFlag (
    ADC_Type *const Base,
    const Adc_SequenceType Seq )
```

Get regular group EOC flag.

**Note**

Function ID: DES\_ADC\_API\_551  
Service ID: none

**Parameters**

<i>in</i>	<i>Base</i>	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
<i>in</i>	<i>Seq</i>	adc sequence

**Returns**

EOC flag of particular sequence

Definition at line 568 of file AC784xx\_Adc\_Reg.h.

**4.1.3.20 Adc\_Reg\_GetRegularEOCInterruptEnableFlag()**

```
LOCAL_INLINE boolean Adc_Reg_GetRegularEOCInterruptEnableFlag (  
    ADC_Type *const Base,  
    const Adc\_SequenceType Seq )
```

Get regular group EOC interrupt enable flag.

**Note**

Function ID: DES\_ADC\_API\_550  
Service ID: none

**Parameters**

<i>in</i>	<i>Base</i>	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
<i>in</i>	<i>Seq</i>	adc sequence

**Returns**

whether EOC interrupt of particular sequence is enabled

Definition at line 552 of file AC784xx\_Adc\_Reg.h.



#### 4.1.3.21 Adc\_Reg\_GetRegularLength()

```
LOCAL_INLINE uint8 Adc_Reg_GetRegularLength (
    ADC_Type *const Base )
```

Get regular group conversion length.

##### Note

Function ID: DES\_ADC\_API\_538  
Service ID: none

##### Parameters

in	Base	adc module
		<ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>

##### Returns

regular group length setting

Definition at line 379 of file AC784xx\_Adc\_Reg.h.

#### 4.1.3.22 Adc\_Reg\_GetRegularParityVal()

```
LOCAL_INLINE uint8 Adc_Reg_GetRegularParityVal (
    ADC_Type *const Base,
    const Adc_SequenceType Seq )
```

Get ADC regular group parity val.

##### Note

Function ID: DES\_ADC\_API\_561  
Service ID: none

##### Parameters

in	Base	adc module
		<ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	Seq	adc sequence

##### Returns

ADC parity val of particular sequence

Definition at line 738 of file AC784xx\_Adc\_Reg.h.

#### 4.1.3.23 Adc\_Reg\_GetSTRFlag()

```
LOCAL_INLINE uint32 Adc_Reg_GetSTRFlag (
    ADC_Type *const Base )
```

Get STR reg flag.

##### Note

Function ID: DES\_ADC\_API\_609  
Service ID: none

##### Parameters

in	Base	adc module
		<ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>

##### Returns

STR flag State

Definition at line 882 of file AC784xx\_Adc\_Reg.h.

#### 4.1.3.24 Adc\_Reg\_SetAM0InjectMode()

```
LOCAL_INLINE void Adc_Reg_SetAM0InjectMode (
    ADC_Type *const Base,
    const boolean State )
```

Set injected group analog monitor mode.

##### Parameters

in	Base	adc module
		<ul style="list-style-type: none"><li>• ADC0</li></ul>
in	State	enabling State
		<ul style="list-style-type: none"><li>• ENABLE</li><li>• DISABLE</li></ul>

**Returns**

none

Definition at line 979 of file AC784xx\_Adc\_Reg.h.

**4.1.3.25 Adc\_Reg\_SetAMOIInterrupt()**

```
LOCAL_INLINE void Adc_Reg_SetAMOIInterrupt (
    ADC_Type *const Base,
    const boolean State )
```

Set AMO Interrupt.

**Parameters**

in	<i>Base</i>	adc module • ADC0
in	<i>State</i>	enabling State • ENABLE • DISABLE

**Returns**

none

Definition at line 917 of file AC784xx\_Adc\_Reg.h.

**4.1.3.26 Adc\_Reg\_SetAMOOffset()**

```
LOCAL_INLINE void Adc_Reg_SetAMOOffset (
    ADC_Type *const Base,
    uint16 HighOffset,
    uint16 LowOffset )
```

Set analog monitor offset.

**Parameters**

in	<i>Base</i>	adc module • ADC0
in	<i>HighOffset</i>	high offset value • 0~0xfff
in	<i>LowOffset</i>	low offset value • 0~0xfff

**Returns**

none

Definition at line 1027 of file AC784xx\_Adc\_Reg.h.

**4.1.3.27 Adc\_Reg\_SetAMORegularMode()**

```
LOCAL_INLINE void Adc_Reg_SetAMORegularMode (
    ADC_Type *const Base,
    const boolean State )
```

Set regular group analog monitor mode.

**Parameters**

in	<i>Base</i>	adc module <ul style="list-style-type: none"> <li>• ADC0</li> </ul>
in	<i>State</i>	enabling State <ul style="list-style-type: none"> <li>• ENABLE</li> <li>• DISABLE</li> </ul>

**Returns**

none

Definition at line 964 of file AC784xx\_Adc\_Reg.h.

**4.1.3.28 Adc\_Reg\_SetAMOSingleChannel()**

```
LOCAL_INLINE void Adc_Reg_SetAMOSingleChannel (
    ADC_Type *const Base,
    const Adc\_InputChannelType Channel )
```

Set analog monitor single channel.

**Parameters**

in	<i>Base</i>	adc module <ul style="list-style-type: none"> <li>• ADC0</li> </ul>
in	<i>Channel</i>	analog monitor channel <ul style="list-style-type: none"> <li>• ADC_CH_0</li> <li>• ADC_CH_1 ...</li> <li>• ADC_CH_TSSENSOR</li> </ul>

**Returns**

none

Definition at line 934 of file AC784xx\_Adc\_Reg.h.

**4.1.3.29 Adc\_Reg\_SetAMOSingleChannelMode()**

```
LOCAL_INLINE void Adc_Reg_SetAMOSingleChannelMode (
    ADC_Type *const Base,
    const boolean State )
```

Set analog monitor single channel mode.

**Parameters**

in	<i>Base</i>	adc module <ul style="list-style-type: none"> <li>• ADC0</li> </ul>
in	<i>State</i>	enabling State <ul style="list-style-type: none"> <li>• ENABLE</li> <li>• DISABLE</li> </ul>

**Returns**

none

Definition at line 949 of file AC784xx\_Adc\_Reg.h.

**4.1.3.30 Adc\_Reg\_SetAMThreshold()**

```
LOCAL_INLINE void Adc_Reg_SetAMThreshold (
    ADC_Type *const Base,
    uint16 HighValue,
    uint16 LowValue )
```

Set analog monitor threshold.

**Parameters**

in	<i>Base</i>	adc module <ul style="list-style-type: none"> <li>• ADC0</li> </ul>
in	<i>HighValue</i>	high threshold value <ul style="list-style-type: none"> <li>• 0~0xff</li> </ul>
in	<i>LowValue</i>	low threshold value <ul style="list-style-type: none"> <li>• 0~0xff</li> </ul>

**Returns**

none

Definition at line 1010 of file AC784xx\_Adc\_Reg.h.

**4.1.3.31 Adc\_Reg\_SetAMOTriggerMode()**

```
LOCAL_INLINE void Adc_Reg_SetAMOTriggerMode (
    ADC_Type *const Base,
    Adc_AmoTriggerModeType Mode )
```

Set analog monitor trigger type.

**Parameters**

in	<i>Base</i>	adc module <ul style="list-style-type: none"><li>• ADC0</li></ul>
in	<i>Mode</i>	amo trigger mode <ul style="list-style-type: none"><li>• ADC_AMO_TRIGGER_LEVEL</li><li>• ADC_AMO_TRIGGER_EDGE</li></ul>

**Returns**

none

Definition at line 994 of file AC784xx\_Adc\_Reg.h.

**4.1.3.32 Adc\_Reg\_SetAverageEnableFlag()**

```
LOCAL_INLINE void Adc_Reg_SetAverageEnableFlag (
    ADC_Type *const Base,
    boolean State )
```

Set hardware average enable flag.

**Parameters**

in	<i>Base</i>	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	<i>State</i>	enable or disable hardware average function

Returns

none

Definition at line 783 of file AC784xx\_Adc\_Reg.h.

4.1.3.33 Adc\_Reg\_SetAverageMode()

```
LOCAL_INLINE void Adc_Reg_SetAverageMode (
    ADC_Type *const Base,
    uint8 averageMode )
```

Set hardware average mode.

Parameters

in	Base	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	averageMode	average mode <ul style="list-style-type: none"><li>• ADC_AVERAGE_4</li><li>• ADC_AVERAGE_8</li><li>• ADC_AVERAGE_16</li><li>• ADC_AVERAGE_32</li></ul>

Returns

none

Definition at line 815 of file AC784xx\_Adc\_Reg.h.

4.1.3.34 Adc\_Reg\_SetCalibrationEnableFlag()

```
LOCAL_INLINE void Adc_Reg_SetCalibrationEnableFlag (
    ADC_Type *const Base,
    boolean State )
```

Set Calibration function enable flag.

Parameters

in	Base	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	State	enable or disbale calibration function

Returns

none

Definition at line 1042 of file AC784xx\_Adc\_Reg.h.

4.1.3.35 Adc\_Reg\_SetChannelSampleTime()

```
LOCAL_INLINE void Adc_Reg_SetChannelSampleTime (
    ADC_Type *const Base,
    const Adc_InputChannelType Channel,
    const Adc_SamplingTimeType Time )
```

Set Channel sample time.

Note

Function ID: DES\_ADC\_API\_541  
Service ID: none

Parameters

in	Base	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	Channel	adc Channel
in	Time	sample time <ul style="list-style-type: none"><li>• ADC_SPT_CLK_5</li><li>• ADC_SPT_CLK_10</li><li>• ADC_SPT_CLK_15</li><li>• ADC_SPT_CLK_23</li><li>• ADC_SPT_CLK_35</li><li>• ADC_SPT_CLK_45</li><li>• ADC_SPT_CLK_85</li><li>• ADC_SPT_CLK_185</li></ul>

Returns

none

Definition at line 434 of file AC784xx\_Adc\_Reg.h.



4.1.3.36 Adc\_Reg\_SetClockPrescaler()

```
LOCAL_INLINE void Adc_Reg_SetClockPrescaler (
    ADC_Type *const Base,
    const Adc_PrescaleType psc )
```

Set adc clock prescaler.

Note

Function ID: DES\_ADC\_API\_507  
Service ID: none

Parameters

in	Base	adc module <ul style="list-style-type: none"><li>• 0: ADC0</li><li>• 1: ADC1</li></ul>
in	psc	prescaler Value <ul style="list-style-type: none"><li>• ADC_CLK_DIVIDE_1</li><li>• ADC_CLK_DIVIDE_2</li><li>• ADC_CLK_DIVIDE_3</li><li>• ADC_CLK_DIVIDE_4</li><li>• ADC_CLK_DIVIDE_5</li><li>• ADC_CLK_DIVIDE_6</li><li>• ADC_CLK_DIVIDE_7</li><li>• ADC_CLK_DIVIDE_8</li><li>• ADC_CLK_DIVIDE_9</li><li>• ADC_CLK_DIVIDE_10</li><li>• ADC_CLK_DIVIDE_11</li><li>• ADC_CLK_DIVIDE_12</li><li>• ADC_CLK_DIVIDE_13</li><li>• ADC_CLK_DIVIDE_14</li><li>• ADC_CLK_DIVIDE_15</li><li>• ADC_CLK_DIVIDE_16</li></ul>

Returns

none

Definition at line 85 of file AC784xx\_Adc\_Reg.h.

#### 4.1.3.37 Adc\_Reg\_SetContinuousEnableFlag()

```
LOCAL_INLINE void Adc_Reg_SetContinuousEnableFlag (
    ADC_Type *const Base,
    const boolean State )
```

Set continuous convert mode.

##### Note

Function ID: DES\_ADC\_API\_519

Service ID: none

##### Parameters

in	<i>Base</i>	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	<i>State</i>	enable or disable continuous convert mode

##### Returns

none

Definition at line 188 of file AC784xx\_Adc\_Reg.h.

#### 4.1.3.38 Adc\_Reg\_SetDataAlign()

```
LOCAL_INLINE void Adc_Reg_SetDataAlign (
    ADC_Type *const Base,
    const Adc_ResultAlignmentType align )
```

Set ADC data Alignment.

##### Note

Function ID: DES\_ADC\_API\_559

Service ID: none

##### Parameters

in	<i>Base</i>	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	<i>align</i>	data Alignment setting <ul style="list-style-type: none"><li>• ADC_ALIGN_RIGHT</li><li>• ADC_ALIGN_LEFT</li></ul>

Returns

none

Definition at line 708 of file AC784xx\_Adc\_Reg.h.

4.1.3.39 Adc\_Reg\_SetDMAEnableFlag()

```
LOCAL_INLINE void Adc_Reg_SetDMAEnableFlag (
    ADC_Type *const Base,
    boolean State )
```

Set ADC DMA enable flag.

Note

Function ID: DES\_ADC\_API\_545  
Service ID: none

Parameters

in	Base	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	State	enable or disbale DMA function in ADC

Returns

none

Definition at line 477 of file AC784xx\_Adc\_Reg.h.

4.1.3.40 Adc\_Reg\_SetGainOffset0Value()

```
LOCAL_INLINE void Adc_Reg_SetGainOffset0Value (
    ADC_Type *const Base,
    sint16 Gain,
    sint16 Offset )
```

Set gain error and offset error value0 for external channel.

Parameters

in	Base	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	Gain	gain error in LSB. Can only be in [-4096, 4095]
in	Offset	offset error in LSB. Can only be in [-1024, 1023]

**Returns**

none

Definition at line 1197 of file AC784xx\_Adc\_Reg.h.

**4.1.3.41 Adc\_Reg\_SetGainOffset1Value()**

```
LOCAL_INLINE void Adc_Reg_SetGainOffset1Value (
    ADC_Type *const Base,
    sint16 Gain,
    sint16 Offset )
```

Set gain error and offset error value0 for internal channel.

**Parameters**

in	<i>Base</i>	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	<i>Gain</i>	gain error in LSB. Can only be in [-4096, 4095]
in	<i>Offset</i>	offset error in LSB. Can only be in [-1024, 1023]

**Returns**

none

Definition at line 1230 of file AC784xx\_Adc\_Reg.h.

**4.1.3.42 Adc\_Reg\_SetGEOEVIN()**

```
LOCAL_INLINE void Adc_Reg_SetGEOEVIN (
    uint8 Instance,
    uint8 Vin )
```

Set GE OE calibration input signal.

**Parameters**

in	<i>Instance</i>	adc instance number
in	<i>Vin</i>	calibration input signal

**Returns**

none

Definition at line 1124 of file AC784xx\_Adc\_Reg.h.

#### 4.1.3.43 Adc\_Reg\_SetGEOEVINEnableFlag()

```
LOCAL_INLINE void Adc_Reg_SetGEOEVINEnableFlag (
    uint8 Instance,
    boolean State )
```

Set GE OE calibration voltage input signal enable flag.

##### Parameters

in	<i>Instance</i>	adc instance number
in	<i>State</i>	enable or disable calibration voltage input

##### Returns

none

Definition at line 1054 of file AC784xx\_Adc\_Reg.h.

#### 4.1.3.44 Adc\_Reg\_SetInjectAutoEnableFlag()

```
LOCAL_INLINE void Adc_Reg_SetInjectAutoEnableFlag (
    ADC_Type *const Base,
    const boolean State )
```

Set ADC automatic inject injection group convert after regular group function enable flag.

##### Note

Function ID: DES\_ADC\_API\_525  
Service ID: none

##### Parameters

in	<i>Base</i>	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	<i>State</i>	enable or disable automatic inject convert mode

##### Returns

none

Definition at line 238 of file AC784xx\_Adc\_Reg.h.

#### 4.1.3.45 Adc\_Reg\_SetInjectConversionChannel()

```
LOCAL_INLINE void Adc_Reg_SetInjectConversionChannel (
    ADC_Type *const Base,
    const Adc_SequenceType Seq,
    Adc_InputChannelType Channel )
```

Set injection group sequence.

##### Note

Function ID: DES\_ADC\_API\_553  
Service ID: none

##### Parameters

in	<i>Base</i>	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	<i>Seq</i>	adc sequence
in	<i>Channel</i>	adc Channel

##### Returns

none

Definition at line 609 of file AC784xx\_Adc\_Reg.h.

#### 4.1.3.46 Adc\_Reg\_SetInjectDiscontinuousEnableFlag()

```
LOCAL_INLINE void Adc_Reg_SetInjectDiscontinuousEnableFlag (
    ADC_Type *const Base,
    const boolean State )
```

Set discontinuous mode for injection group.

##### Note

Function ID: DES\_ADC\_API\_527  
Service ID: none

##### Parameters

in	<i>Base</i>	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	<i>State</i>	enable or disable injection group discontinuous convert mode

**Returns**

none

Definition at line 255 of file AC784xx\_Adc\_Reg.h.

**4.1.3.47 Adc\_Reg\_SetInjectEOCInterruptEnableFlag()**

```
LOCAL_INLINE void Adc_Reg_SetInjectEOCInterruptEnableFlag (
    ADC_Type *const Base,
    const Adc_SequenceType Seq,
    const boolean State )
```

Set injection group end of conversion(EOC) interrupt enable flag.

**Note**

Function ID: DES\_ADC\_API\_555

Service ID: none

**Parameters**

in	<i>Base</i>	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	<i>Seq</i>	adc sequence
in	<i>State</i>	enable or disbale EOC interrupt

**Returns**

none

Definition at line 630 of file AC784xx\_Adc\_Reg.h.

**4.1.3.48 Adc\_Reg\_SetInjectLength()**

```
LOCAL_INLINE void Adc_Reg_SetInjectLength (
    ADC_Type *const Base,
    const uint8 length )
```

Set injection group conversion length.

**Note**

Function ID: DES\_ADC\_API\_539

Service ID: none

## Parameters

in	<i>Base</i>	adc module <ul style="list-style-type: none"> <li>• ADC0</li> <li>• ADC1</li> </ul>
in	<i>length</i>	injection group length

## Returns

none

Definition at line 394 of file AC784xx\_Adc\_Reg.h.

## 4.1.3.49 Adc\_Reg\_SetInjectOffset()

```
LOCAL_INLINE void Adc_Reg_SetInjectOffset (
    ADC_Type *const Base,
    uint8 Num,
    uint16 Offset )
```

Set injection group offset Value.

## Parameters

in	<i>Base</i>	adc module <ul style="list-style-type: none"> <li>• ADC0</li> <li>• ADC1</li> </ul>
in	<i>Num</i>	injection group index, should be less than ADC_INJECT_SEQUENCE_LEGNTH
in	<i>Offset</i>	offset value, which will be subtracted after injection group index conversion complete.

## Returns

none

Definition at line 335 of file AC784xx\_Adc\_Reg.h.

## 4.1.3.50 Adc\_Reg\_SetInjectTriggerSource()

```
LOCAL_INLINE void Adc_Reg_SetInjectTriggerSource (
    ADC_Type *const Base,
    const Adc\_TriggerSourceType source )
```

Set injected group trigger source.

## Note

Function ID: DES\_ADC\_API\_532  
Service ID: none



## Parameters

in	<i>Base</i>	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	<i>source</i>	trigger source type <ul style="list-style-type: none"><li>• ADC_TRIGG_SRC_SW</li><li>• ADC_TRIGG_SRC_HW</li></ul>

## Returns

none

Definition at line 319 of file AC784xx\_Adc\_Reg.h.

## 4.1.3.51 Adc\_Reg\_SetIntervalEnableFlag()

```
LOCAL_INLINE void Adc_Reg_SetIntervalEnableFlag (  
    ADC_Type *const Base,  
    const boolean State )
```

Set ADC interval mode.

## Note

Function ID: DES\_ADC\_API\_517

Service ID: none Can only be configured in mode3/5, other modes do not work.

## Parameters

in	<i>Base</i>	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	<i>State</i>	enable or disable scan convert mode

## Returns

none

Definition at line 171 of file AC784xx\_Adc\_Reg.h.

#### 4.1.3.52 Adc\_Reg\_SetPowerEnableFlag()

```
LOCAL_INLINE void Adc_Reg_SetPowerEnableFlag (
    ADC_Type *const Base,
    const boolean mode )
```

Set adc power mode.

##### Note

Function ID: DES\_ADC\_API\_513

Service ID: none

##### Parameters

in	<i>Base</i>	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	<i>mode</i>	enable or disable ADC power

##### Returns

none

Definition at line 135 of file AC784xx\_Adc\_Reg.h.

#### 4.1.3.53 Adc\_Reg\_SetRegularConversionChannel()

```
LOCAL_INLINE void Adc_Reg_SetRegularConversionChannel (
    ADC_Type *const Base,
    const Adc_SequenceType Seq,
    Adc_InputChannelType Channel )
```

Set regular group sequence.

##### Note

Function ID: DES\_ADC\_API\_547

Service ID: none

##### Parameters

in	<i>Base</i>	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	<i>Seq</i>	adc sequence
in	<i>Channel</i>	adc Channel

**Returns**

none

Definition at line 510 of file AC784xx\_Adc\_Reg.h.

**4.1.3.54 Adc\_Reg\_SetRegularDiscontinuousEnableFlag()**

```
LOCAL_INLINE void Adc_Reg_SetRegularDiscontinuousEnableFlag (  
    ADC_Type *const Base,  
    const boolean State )
```

Set discontinuous mode enable flag of regular group.

**Note**

Function ID: DES\_ADC\_API\_521

Service ID: none

**Parameters**

in	<i>Base</i>	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	<i>State</i>	enable or disable discontinuous convert mode

**Returns**

none

Definition at line 205 of file AC784xx\_Adc\_Reg.h.

**4.1.3.55 Adc\_Reg\_SetRegularDiscontinuousNum()**

```
LOCAL_INLINE void Adc_Reg_SetRegularDiscontinuousNum (  
    ADC_Type *const Base,  
    const uint8 num )
```

Set discontinuous mode number for regular group.

**Note**

Function ID: DES\_ADC\_API\_523

Service ID: none

## Parameters

in	<i>Base</i>	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	<i>num</i>	discontinuous convert number for regular group

## Returns

none

Definition at line 222 of file AC784xx\_Adc\_Reg.h.

## 4.1.3.56 Adc\_Reg\_SetRegularEOCInterruptEnableFlag()

```
LOCAL_INLINE void Adc_Reg_SetRegularEOCInterruptEnableFlag (  
    ADC_Type *const Base,  
    const Adc_SequenceType Seq,  
    const boolean State )
```

Set regular group end of conversion(EOC) interrupt enable flag.

## Note

Function ID: DES\_ADC\_API\_549  
Service ID: none

## Parameters

in	<i>Base</i>	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	<i>Seq</i>	adc sequence
in	<i>State</i>	enable or disbale EOC interrupt

## Returns

none

Definition at line 531 of file AC784xx\_Adc\_Reg.h.

## 4.1.3.57 Adc\_Reg\_SetRegularLength()

```
LOCAL_INLINE void Adc_Reg_SetRegularLength (  
    ADC_Type *const Base,  
    const uint8 length )
```

Set regualr group conversion length.

**Note**

Function ID: DES\_ADC\_API\_537  
Service ID: none

**Parameters**

in	<i>Base</i>	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	<i>length</i>	regular group length

**Returns**

none

Definition at line 364 of file AC784xx\_Adc\_Reg.h.

**4.1.3.58 Adc\_Reg\_SetRegularTriggerSource()**

```
LOCAL_INLINE void Adc_Reg_SetRegularTriggerSource (  
    ADC_Type *const Base,  
    const Adc\_TriggerSourceType source )
```

Set regular group trigger source.

**Note**

Function ID: DES\_ADC\_API\_529  
Service ID: none

**Parameters**

in	<i>Base</i>	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	<i>source</i>	trigger source type <ul style="list-style-type: none"><li>• ADC_TRIGG_SRC_SW</li><li>• ADC_TRIGG_SRC_HW</li></ul>

**Returns**

none

Definition at line 274 of file AC784xx\_Adc\_Reg.h.

4.1.3.59 Adc\_Reg\_SetResolution()

```
LOCAL_INLINE void Adc_Reg_SetResolution (
    ADC_Type *const Base,
    const Adc_ResolutionType Resolution )
```

Set adc Resolution.

Note

Function ID: DES\_ADC\_API\_509  
Service ID: none

Parameters

in	Base	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	Resolution	Resolution setting <ul style="list-style-type: none"><li>• ADC_RESOLUTION_12↔BIT</li><li>• ADC_RESOLUTION_10↔BIT</li><li>• ADC_RESOLUTION_8BIT</li></ul>

Returns

none

Definition at line 103 of file AC784xx\_Adc\_Reg.h.

4.1.3.60 Adc\_Reg\_SetScanEnableFlag()

```
LOCAL_INLINE void Adc_Reg_SetScanEnableFlag (
    ADC_Type *const Base,
    const boolean State )
```

Set scan convert mode enable flag.

Note

Function ID: DES\_ADC\_API\_515  
Service ID: none

## Parameters

in	<i>Base</i>	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	<i>State</i>	enable or disable scan convert mode

## Returns

none

Definition at line 152 of file AC784xx\_Adc\_Reg.h.

## 4.1.3.61 Adc\_Reg\_SetVoltageReference()

```
LOCAL_INLINE void Adc_Reg_SetVoltageReference (
    ADC_Type *const Base,
    const Adc\_VoltageReferenceType VoltageRef )
```

Set adc voltage reference source.

## Note

Function ID: DES\_ADC\_API\_511  
Service ID: none

## Parameters

in	<i>Base</i>	adc module <ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>
in	<i>VoltageRef</i>	voltage reference source <ul style="list-style-type: none"><li>• ADC_VOLTAGEREF_VREF</li><li>• ADC_VOLTAGEREF_VDDA</li></ul>

## Returns

none

Definition at line 120 of file AC784xx\_Adc\_Reg.h.

#### 4.1.3.62 Adc\_Reg\_SoftwareStartInjectConvert()

```
LOCAL_INLINE void Adc_Reg_SoftwareStartInjectConvert (
    ADC_Type *const Base )
```

Start software trigger for inject group convert.

##### Note

Function ID: DES\_ADC\_API\_531  
Service ID: none

##### Parameters

in	Base	adc module
		<ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>

##### Returns

none

Definition at line 302 of file AC784xx\_Adc\_Reg.h.

#### 4.1.3.63 Adc\_Reg\_SoftwareStartRegularConvert()

```
LOCAL_INLINE void Adc_Reg_SoftwareStartRegularConvert (
    ADC_Type *const Base )
```

Start software trigger for regular group convert.

##### Note

Function ID: DES\_ADC\_API\_531  
Service ID: none

##### Parameters

in	Base	adc module
		<ul style="list-style-type: none"><li>• ADC0</li><li>• ADC1</li></ul>

##### Returns

none

Definition at line 288 of file AC784xx\_Adc\_Reg.h.



## 4.1.3.64 Adc\_Reg\_SwRstClkCtrl()

```
LOCAL_INLINE void Adc_Reg_SwRstClkCtrl (
    uint32 Instance,
    boolean IsEnable )
```

Adc\_Reg\_SwRstClkCtrl.

## Note

Function ID: DES\_ADC\_API\_606  
Service ID: none

## Parameters

in	<i>Instance</i>	adc module instance
in	<i>IsEnable</i>	enable/disable

## Returns

none

Definition at line 845 of file AC784xx\_Adc\_Reg.h.

## 4.2 AC784xx\_API\_Reference\_Manual\_ADC.pdf File Reference

## 4.3 Adc\_Hal.c File Reference

This file provides all adc hal api.

```
#include "AC784xx_Adc_Reg.h"
#include "AC784xx_Ctu_Reg.h"
#include "OsIf_Time.h"
#include "Ckgen_Hal.h"
#include "Rcm_Hal.h"
#include "Dma_Hal.h"
#include "Adc_Hal.h"
#include "Core_Hal.h"
```

## Classes

- struct [adc\\_calibration\\_t](#)  
ADC calibration structure.

## Macros

- #define [ROUND](#)(x) ((x) >= 0 ? ((sint16)((x) + 0.5)) : ((sint16)((x) - 0.5)))

## Enumerations

- enum [Ana\\_GeoecalVinType](#) {  
[ANA\\_GEOECAL\\_FLOATING](#) = 0U, [ANA\\_GEOECAL\\_REF\\_16\\_4](#), [ANA\\_GEOECAL\\_REF\\_16\\_5](#), [ANA\\_GEOECAL\\_REF\\_16\\_6](#),  
[ANA\\_GEOECAL\\_REF\\_16\\_8](#), [ANA\\_GEOECAL\\_REF\\_16\\_10](#), [ANA\\_GEOECAL\\_REF\\_16\\_11](#), [ANA\\_GEOECAL\\_REF\\_16\\_12](#) }

*Analog ADC GE OE calibration internal input source.*

## Functions

- void [Adc\\_Hal\\_Init](#) (uint8 Instance, const [Adc\\_InitConfigType](#) \*Config)  
*init ADC module about [Adc\\_InitConfigType](#) struct to hw*
- void [Adc\\_Hal\\_Deinit](#) (uint8 Instance)  
*Deinitialize ADC module hw.*
- void [Adc\\_Hal\\_InitConverterStruct](#) ([Adc\\_ConverterConfigType](#) \*const Config)  
*Initialize converter config struct default args.*
- void [Adc\\_Hal\\_ConfigConverter](#) (uint8 Instance, const [Adc\\_ConverterConfigType](#) \*Config)  
*adc Config convert args to hw*
- void [Adc\\_Hal\\_InitChanStruct](#) ([Adc\\_ChanConfigType](#) \*const Config)  
*Initialize adc channel config struct default args.*
- void [Adc\\_Hal\\_ConfigChannel](#) (uint8 Instance, const [Adc\\_ChanConfigType](#) \*Config)  
*adc config the Channel args to the hw.*
- void [Adc\\_Hal\\_ConfigRegularGroup](#) (uint32 Instance, [Adc\\_GroupConfigType](#) \*Config)  
*: Apply the Regular Group Mode configuration to the hardware.*
- void [Adc\\_Hal\\_ConfigInjectGroup](#) (uint32 Instance, [Adc\\_GroupConfigType](#) \*Config)  
*: Apply the Inject Group Mode configuration to the hardware.*
- void [Adc\\_Hal\\_DmaEnable](#) (uint32 Instance, boolean Enable)  
*: Enable adc-dma control.*
- void [Adc\\_Hal\\_SwTriggerRegularConvert](#) (uint8 Instance)  
*Start software trigger regular group conversion.*
- void [Adc\\_Hal\\_SwTriggerInjectConvert](#) (uint8 Instance)  
*Start software trigger inject group conversion.*
- uint16 [Adc\\_Hal\\_GetSeqResult](#) (uint8 Instance, [Adc\\_SequenceType](#) SeqIndex)  
*Get the ADC conversion result for the group sequence.*
- void [Adc\\_Hal\\_InitGroupStruct](#) ([Adc\\_GroupConfigType](#) \*const Config)  
*Initialize adc amo config struct default args.*
- void [Adc\\_Hal\\_ConfigGroup](#) (uint8 Instance, const [Adc\\_GroupConfigType](#) \*Config)  
*: Apply the Group Mode configuration to the hardware.*
- void [Adc\\_Hal\\_InitAMOSTruct](#) ([Adc\\_AmoConfigType](#) \*const Config)  
*initialize Analog Monitor config structure*
- void [Adc\\_Hal\\_ConfigAmo](#) (uint8 Instance, const [Adc\\_AmoConfigType](#) \*Config)  
*adc config the amo args to the hw.*
- boolean [Adc\\_Hal\\_GetIdleFlag](#) (uint8 Instance)  
*Get ADC Idle flag.*
- boolean [Adc\\_Hal\\_GetConvertCompleteFlag](#) (uint8 Instance, [Adc\\_SequenceType](#) SeqIndex)  
*Get conversion complete flag.*
- void [Adc\\_Hal\\_ClearConvertCompleteFlag](#) (uint8 Instance, [Adc\\_SequenceType](#) SeqIndex)  
*Clear conversion complete flag.*
- boolean [Adc\\_Hal\\_GetTriggerConflictFlag](#) (uint8 Instance)  
*Get the trigger conflict flags.*
- void [Adc\\_Hal\\_ClearTriggerConflictFlag](#) (uint8 Instance)  
*Clear the trigger conflict flags.*

- uint8 [Adc\\_Hal\\_GetParityVal](#) (uint8 Instance, [Adc\\_SequenceType](#) SeqIndex)  
*Get ADC group sequeunce index parity value.*
- float32 [Adc\\_Hal\\_ConvertToTemperature](#) (const uint16 Value, const float RefVoltage)  
*Convert ADC value of T-Sensor to Temperature.*
- void [Adc\\_Hal\\_SetInjectOffset](#) (uint8 Instance, [Adc\\_SequenceType](#) SeqIndex, uint16 Offset)  
*Set the adc inject group offset.*
- void [Adc\\_Hal\\_ConfigAutoCalibration](#) (uint8 Instance)  
*Config autocalibration.*
- ADC\_Type \* [Adc\\_Hal\\_GetBase](#) (uint8 Instance)  
*Get the adc base address.*
- [ISR](#) (ADC0\_IRQHandler)  
*ADC0 Irq.*
- [ISR](#) (ADC1\_IRQHandler)  
*ADC1 Irq.*

4.3.1 Detailed Description

This file provides all adc hal api.

4.3.2 Macro Definition Documentation

4.3.2.1 ROUND

```
#define ROUND(  
    x ) ((x) >= 0 ? ((sint16)((x) + 0.5)) : ((sint16)((x) - 0.5)))
```

Definition at line 48 of file Adc\_Hal.c.

4.3.3 Enumeration Type Documentation

4.3.3.1 Ana\_GeoecalVinType

```
enum Ana\_GeoecalVinType
```

Analog ADC GE OE calibration internal input source.

Enumerator

ANA_GEOECAL_FLOATING	
ANA_GEOECAL_REF_16_4	
ANA_GEOECAL_REF_16_5	
ANA_GEOECAL_REF_16_6	
ANA_GEOECAL_REF_16_8	
ANA_GEOECAL_REF_16_10	
ANA_GEOECAL_REF_16_11	
ANA_GEOECAL_REF_16_12	

Definition at line 53 of file Adc\_Hal.c.

#### 4.3.4 Function Documentation

##### 4.3.4.1 Adc\_Hal\_ClearConvertCompleteFlag()

```
void Adc_Hal_ClearConvertCompleteFlag (
    uint8 Instance,
    Adc_SequenceType SeqIndex )
```

Clear conversion complete flag.

##### Note

Function ID: DES\_ADC\_API\_011  
Service ID: none

##### Parameters

in	<i>Instance</i>	ADC Instance number
in	<i>SeqIndex</i>	ADC group sequence index

##### Returns

void

Definition at line 459 of file Adc\_Hal.c.

##### 4.3.4.2 Adc\_Hal\_ClearTriggerConflictFlag()

```
void Adc_Hal_ClearTriggerConflictFlag (
    uint8 Instance )
```

Clear the trigger conflict flags.

##### Note

Function ID: DES\_ADC\_API\_013  
Service ID: none

##### Parameters

in	<i>Instance</i>	ADC Instance number
----	-----------------	---------------------

**Returns**

void

Definition at line 483 of file Adc\_Hal.c.

**4.3.4.3 Adc\_Hal\_ConfigAmo()**

```
void Adc_Hal_ConfigAmo (
    uint8 Instance,
    const Adc_AmoConfigType * Config )
```

adc config the amo args to the hw.

**Note**

Function ID: DES\_ADC\_API\_005

Service ID: none

**Parameters**

in	<i>Instance</i>	ADC Instance number
in	<i>Config</i>	configuration that need to apply

**Returns**

void

Definition at line 415 of file Adc\_Hal.c.

**4.3.4.4 Adc\_Hal\_ConfigAutoCalibration()**

```
void Adc_Hal_ConfigAutoCalibration (
    uint8 Instance )
```

Config autocalibration.

**Note**

Function ID: DES\_ADC\_API\_023

Service ID: none

**Parameters**

in	<i>Instance</i>	ADC Instance number
----	-----------------	---------------------

**Returns**

adc base address ptr

Definition at line 528 of file Adc\_Hal.c.

**4.3.4.5 Adc\_Hal\_ConfigChannel()**

```
void Adc_Hal_ConfigChannel (
    uint8 Instance,
    const Adc_ChainConfigType * Config )
```

adc config the Channel args to the hw.

**Note**

Function ID: DES\_ADC\_API\_003

Service ID: none

**Parameters**

in	<i>Instance</i>	ADC Instance number
in	<i>Config</i>	configuration that need to apply

**Returns**

void

Definition at line 240 of file Adc\_Hal.c.

**4.3.4.6 Adc\_Hal\_ConfigConverter()**

```
void Adc_Hal_ConfigConverter (
    uint8 Instance,
    const Adc_ConverterConfigType * Config )
```

adc Config convert args to hw

**Note**

Function ID: DES\_ADC\_API\_002

Service ID: none

**Parameters**

in	<i>Instance</i>	ADC Instance number
in	<i>Config</i>	convert Config that need to Config

**Returns**

void

Definition at line 202 of file Adc\_Hal.c.

**4.3.4.7 Adc\_Hal\_ConfigGroup()**

```
void Adc_Hal_ConfigGroup (
    uint8 Instance,
    const Adc_GroupConfigType * Config )
```

: Apply the Group Mode configuration to the hardware.

**Note**

Function ID: DES\_ADC\_API\_004

Service ID: none

**Parameters**

in	<i>Instance</i>	ADC Instance number
in	<i>Config</i>	configuration that need to apply to the group

**Returns**

: void

Definition at line 367 of file Adc\_Hal.c.

**4.3.4.8 Adc\_Hal\_ConfigInjectGroup()**

```
void Adc_Hal_ConfigInjectGroup (
    uint32 Instance,
    Adc_GroupConfigType * Config )
```

: Apply the Inject Group Mode configuration to the hardware.

**Note**

Function ID: DES\_ADC\_API\_19

Service ID: none

**Parameters**

in	<i>Instance</i>	ADC Instance number
in	<i>Config</i>	configuration that need to apply to the group

**Returns**

: void

Definition at line 294 of file Adc\_Hal.c.

**4.3.4.9 Adc\_Hal\_ConfigRegularGroup()**

```
void Adc_Hal_ConfigRegularGroup (
    uint32 Instance,
    Adc_GroupConfigType * Config )
```

: Apply the Regular Group Mode configuration to the hardware.

**Note**

Function ID: DES\_ADC\_API\_18

Service ID: none

**Parameters**

in	<i>Instance</i>	ADC Instance number
in	<i>Config</i>	configuration that need to apply to the group

**Returns**

: void

Definition at line 284 of file Adc\_Hal.c.

**4.3.4.10 Adc\_Hal\_ConvertToTemperature()**

```
float32 Adc_Hal_ConvertToTemperature (
    const uint16 Value,
    const float RefVoltage )
```

Convert ADC value of T-Sensor to Temperature.

**Note**

Function ID: DES\_ADC\_API\_025

Service ID: none

**Parameters**

in	<i>Value</i>	ADC conversion result of internal T-Sensor
in	<i>RefVoltage</i>	reference voltage of ADC, in mV



**Returns**

temperature value in celsius scale

Definition at line 509 of file Adc\_Hal.c.

**4.3.4.11 Adc\_Hal\_Deinit()**

```
void Adc_Hal_Deinit (
    uint8 Instance )
```

Deinitialize ADC module hw.

**Note**

Function ID: DES\_ADC\_API\_001  
Service ID: none

**Parameters**

in	<i>Instance</i>	ADC Instance number
----	-----------------	---------------------

**Returns**

void

Definition at line 175 of file Adc\_Hal.c.

**4.3.4.12 Adc\_Hal\_DmaEnable()**

```
void Adc_Hal_DmaEnable (
    uint32 Instance,
    boolean Enable )
```

: Enable adc-dma control.

**Note**

Function ID: DES\_ADC\_API\_20  
Service ID: none

**Parameters**

in	<i>Instance</i>	ADC Instance number
in	<i>Enable</i>	Enable flag.

**Returns**

: void

Definition at line 304 of file Adc\_Hal.c.

**4.3.4.13 Adc\_Hal\_GetBase()**

```
ADC_Type* Adc_Hal_GetBase (
    uint8 Instance )
```

Get the adc base address.

**Note**

Function ID: DES\_ADC\_API\_022  
Service ID: none

**Parameters**

in	<i>Instance</i>	ADC Instance number
----	-----------------	---------------------

**Returns**

adc base address ptr

Definition at line 593 of file Adc\_Hal.c.

**4.3.4.14 Adc\_Hal\_GetConvertCompleteFlag()**

```
boolean Adc_Hal_GetConvertCompleteFlag (
    uint8 Instance,
    Adc\_SequenceType SeqIndex )
```

Get conversion complete flag.

**Note**

Function ID: DES\_ADC\_API\_010  
Service ID: none

**Parameters**

in	<i>Instance</i>	ADC Instance number
in	<i>SeqIndex</i>	ADC group sequence index

**Returns**

whether ADC group sequence index complete, -TRUE: ADC convert complete -FALSE: ADC convert not complete

Definition at line 440 of file Adc\_Hal.c.

**4.3.4.15 Adc\_Hal\_GetIdleFlag()**

```
boolean Adc_Hal_GetIdleFlag (
    uint8 Instance )
```

Get ADC Idle flag.

**Note**

Function ID: DES\_ADC\_API\_009  
Service ID: none

**Parameters**

in	<i>Instance</i>	ADC Instance number
----	-----------------	---------------------

**Returns**

whether ADC is idle, -TRUE: ADC is not converting -FALSE: ADC is busy converting

Definition at line 433 of file Adc\_Hal.c.

**4.3.4.16 Adc\_Hal\_GetParityVal()**

```
uint8 Adc_Hal_GetParityVal (
    uint8 Instance,
    Adc\_SequenceType SeqIndex )
```

Get ADC group sequence index parity value.

**Note**

Function ID: DES\_ADC\_API\_21  
Service ID: none

**Parameters**

in	<i>Instance</i>	ADC Instance number
in	<i>SeqIndex</i>	ADC group sequence index

**Returns**

ADC group sequence index parity value

Definition at line 491 of file Adc\_Hal.c.

**4.3.4.17 Adc\_Hal\_GetSeqResult()**

```
uint16 Adc_Hal_GetSeqResult (
    uint8 Instance,
    Adc_SequenceType SeqIndex )
```

Get the ADC conversion result for the group sequence.

**Note**

Function ID: DES\_ADC\_API\_008

Service ID: none

**Parameters**

in	<i>Instance</i>	ADC Instance number
in	<i>SeqIndex</i>	ADC group sequence index

**Returns**

ADC conversion result

Definition at line 327 of file Adc\_Hal.c.

**4.3.4.18 Adc\_Hal\_GetTriggerConflictFlag()**

```
boolean Adc_Hal_GetTriggerConflictFlag (
    uint8 Instance )
```

Get the trigger conflict flags.

**Note**

Function ID: DES\_ADC\_API\_012

Service ID: none

**Parameters**

in	<i>Instance</i>	ADC Instance number
----	-----------------	---------------------

**Returns**

whether ADC trigger conflict, -TRUE: ADC trigger conflict -FALSE: ADC trigger not conflict

Definition at line 475 of file Adc\_Hal.c.

**4.3.4.19 Adc\_Hal\_Init()**

```
void Adc_Hal_Init (
    uint8 Instance,
    const Adc_InitConfigType * Config )
```

init ADC module about [Adc\\_InitConfigType](#) struct to hw

**Note**

Function ID: DES\_ADC\_API\_000

Service ID: none

**Parameters**

in	<i>Instance</i>	ADC Instance number
in	<i>Config</i>	ADC init config args

**Returns**

void

Definition at line 163 of file Adc\_Hal.c.

**4.3.4.20 Adc\_Hal\_InitAMOStruct()**

```
void Adc_Hal_InitAMOStruct (
    Adc_AmoConfigType *const Config )
```

initialize Analog Monitor config structure

**Note**

Function ID: DES\_ADC\_API\_017

Service ID: none

**Parameters**

out	<i>Config</i>	pointer to the config that need to be initialized
-----	---------------	---

**Returns**

none

Definition at line 396 of file Adc\_Hal.c.

**4.3.4.21 Adc\_Hal\_InitChanStruct()**

```
void Adc_Hal_InitChanStruct (
    Adc_ChainConfigType *const Config )
```

Initialize adc channel config struct default args.

**Note**

Function ID: DES\_ADC\_API\_016  
Service ID: none

**Parameters**

out	<i>Config</i>	adc channel Config structure that need to be initialized
-----	---------------	--

**Returns**

none

Definition at line 229 of file Adc\_Hal.c.

**4.3.4.22 Adc\_Hal\_InitConverterStruct()**

```
void Adc_Hal_InitConverterStruct (
    Adc_ConverterConfigType *const Config )
```

Initialize converter config struct default args.

**Note**

Function ID: DES\_ADC\_API\_014  
Service ID: none

**Parameters**

out	<i>Config</i>	converter config structure that need to be initialized
-----	---------------	--

**Returns**

none

Definition at line 188 of file Adc\_Hal.c.

#### 4.3.4.23 Adc\_Hal\_InitGroupStruct()

```
void Adc_Hal_InitGroupStruct (
    Adc\_GroupConfigType *const Config )
```

Initialize adc amo config struct default args.

##### Note

Function ID: DES\_ADC\_API\_015  
Service ID: none

##### Parameters

out	<i>Config</i>	adc amo Config structure that need to be initialized
-----	---------------	--

##### Returns

none

Definition at line 347 of file Adc\_Hal.c.

#### 4.3.4.24 Adc\_Hal\_SetInjectOffset()

```
void Adc_Hal_SetInjectOffset (
    uint8 Instance,
    Adc\_SequenceType SeqIndex,
    uint16 Offset )
```

Set the adc inject group offset.

##### Note

Function ID: DES\_ADC\_API\_024  
Service ID: none

##### Parameters

in	<i>Instance</i>	ADC Instance number
in	<i>SeqIndex</i>	ADC group sequeunce index
in	<i>Offset</i>	ADC inject group offset value

##### Returns

void

Definition at line 519 of file Adc\_Hal.c.

#### 4.3.4.25 Adc\_Hal\_SwTriggerInjectConvert()

```
void Adc_Hal_SwTriggerInjectConvert (
    uint8 Instance )
```

Start software trigger inject group conversion.

##### Note

Function ID: DES\_ADC\_API\_007  
Service ID: none

##### Parameters

in	<i>Instance</i>	ADC Instance number
----	-----------------	---------------------

##### Returns

void

Definition at line 319 of file Adc\_Hal.c.

#### 4.3.4.26 Adc\_Hal\_SwTriggerRegularConvert()

```
void Adc_Hal_SwTriggerRegularConvert (
    uint8 Instance )
```

Start software trigger regular group conversion.

##### Note

Function ID: DES\_ADC\_API\_006  
Service ID: none

##### Parameters

in	<i>Instance</i>	ADC Instance number
----	-----------------	---------------------

##### Returns

void

Definition at line 311 of file Adc\_Hal.c.



**4.3.4.27 ISR()** [1/2]

```
ISR (
    ADC0_IRQHandler )
```

ADC0 Irq.

**Returns**

void

Definition at line 944 of file Adc\_Hal.c.

**4.3.4.28 ISR()** [2/2]

```
ISR (
    ADC1_IRQHandler )
```

ADC1 Irq.

**Returns**

void

Definition at line 953 of file Adc\_Hal.c.

**4.4 Adc\_Hal.h File Reference**

This file provides all adc hal api.

```
#include "Adc_Hal_Types.h"
```

**Functions**

- void [Adc\\_Hal\\_Init](#) (uint8 Instance, const [Adc\\_InitConfigType](#) \*Config)  
*init ADC module about [Adc\\_InitConfigType](#) struct to hw*
- void [Adc\\_Hal\\_Deinit](#) (uint8 Instance)  
*Deinitialize ADC module hw.*
- void [Adc\\_Hal\\_ConfigConverter](#) (uint8 Instance, const [Adc\\_ConverterConfigType](#) \*Config)  
*adc Config convert args to hw*
- void [Adc\\_Hal\\_ConfigChannel](#) (uint8 Instance, const [Adc\\_ChanConfigType](#) \*Config)  
*adc config the Channel args to the hw.*
- void [Adc\\_Hal\\_ConfigGroup](#) (uint8 Instance, const [Adc\\_GroupConfigType](#) \*Config)  
*: Apply the Group Mode configuration to the hardware.*
- void [Adc\\_Hal\\_ConfigAmo](#) (uint8 Instance, const [Adc\\_AmoConfigType](#) \*Config)  
*adc config the amo args to the hw.*
- void [Adc\\_Hal\\_SwTriggerRegularConvert](#) (uint8 Instance)  
*Start software trigger regular group conversion.*

- void [Adc\\_Hal\\_SwTriggerInjectConvert](#) (uint8 Instance)  
*Start software trigger inject group conversion.*
- uint16 [Adc\\_Hal\\_GetSeqResult](#) (uint8 Instance, [Adc\\_SequenceType](#) SeqIndex)  
*Get the ADC conversion result for the group sequence.*
- boolean [Adc\\_Hal\\_GetIdleFlag](#) (uint8 Instance)  
*Get ADC Idle flag.*
- boolean [Adc\\_Hal\\_GetConvertCompleteFlag](#) (uint8 Instance, [Adc\\_SequenceType](#) SeqIndex)  
*Get conversion complete flag.*
- void [Adc\\_Hal\\_ClearConvertCompleteFlag](#) (uint8 Instance, [Adc\\_SequenceType](#) SeqIndex)  
*Clear conversion complete flag.*
- boolean [Adc\\_Hal\\_GetTriggerConflictFlag](#) (uint8 Instance)  
*Get the trigger conflict flags.*
- void [Adc\\_Hal\\_ClearTriggerConflictFlag](#) (uint8 Instance)  
*Clear the trigger conflict flags.*
- void [Adc\\_Hal\\_InitConverterStruct](#) ([Adc\\_ConverterConfigType](#) \*const Config)  
*Initialize converter config struct default args.*
- void [Adc\\_Hal\\_InitGroupStruct](#) ([Adc\\_GroupConfigType](#) \*const Config)  
*Initialize adc amo config struct default args.*
- void [Adc\\_Hal\\_InitChanStruct](#) ([Adc\\_ChanConfigType](#) \*const Config)  
*Initialize adc channel config struct default args.*
- void [Adc\\_Hal\\_InitAMOStruct](#) ([Adc\\_AmoConfigType](#) \*const Config)  
*initialize Analog Monitor config structure*
- void [Adc\\_Hal\\_ConfigRegularGroup](#) (uint32 Instance, [Adc\\_GroupConfigType](#) \*Config)  
*: Apply the Regular Group Mode configuration to the hardware.*
- void [Adc\\_Hal\\_ConfigInjectGroup](#) (uint32 Instance, [Adc\\_GroupConfigType](#) \*Config)  
*: Apply the Inject Group Mode configuration to the hardware.*
- void [Adc\\_Hal\\_DmaEnable](#) (uint32 Instance, boolean Enable)  
*: Enable adc-dma control.*
- uint8 [Adc\\_Hal\\_GetParityVal](#) (uint8 Instance, [Adc\\_SequenceType](#) SeqIndex)  
*Get ADC group sequence index parity value.*
- [ADC\\_Type](#) \* [Adc\\_Hal\\_GetBase](#) (uint8 Instance)  
*Get the adc base address.*
- void [Adc\\_Hal\\_ConfigAutoCalibration](#) (uint8 Instance)  
*Config autocalibration.*
- void [Adc\\_Hal\\_SetInjectOffset](#) (uint8 Instance, [Adc\\_SequenceType](#) SeqIndex, uint16 Offset)  
*Set the adc inject group offset.*
- float32 [Adc\\_Hal\\_ConvertToTemperature](#) (const uint16 Value, const float RefVoltage)  
*Convert ADC value of T-Sensor to Temperature.*

#### 4.4.1 Detailed Description

This file provides all adc hal api.

#### 4.4.2 Function Documentation

#### 4.4.2.1 Adc\_Hal\_ClearConvertCompleteFlag()

```
void Adc_Hal_ClearConvertCompleteFlag (
    uint8 Instance,
    Adc_SequenceType SeqIndex )
```

Clear conversion complete flag.

##### Note

Function ID: DES\_ADC\_API\_011  
Service ID: none

##### Parameters

in	<i>Instance</i>	ADC Instance number
in	<i>SeqIndex</i>	ADC group sequence index

##### Returns

void

Definition at line 459 of file Adc\_Hal.c.

#### 4.4.2.2 Adc\_Hal\_ClearTriggerConflictFlag()

```
void Adc_Hal_ClearTriggerConflictFlag (
    uint8 Instance )
```

Clear the trigger conflict flags.

##### Note

Function ID: DES\_ADC\_API\_013  
Service ID: none

##### Parameters

in	<i>Instance</i>	ADC Instance number
----	-----------------	---------------------

##### Returns

void

Definition at line 483 of file Adc\_Hal.c.

#### 4.4.2.3 Adc\_Hal\_ConfigAmo()

```
void Adc_Hal_ConfigAmo (
    uint8 Instance,
    const Adc_AmoConfigType * Config )
```

adc config the amo args to the hw.

##### Note

Function ID: DES\_ADC\_API\_005  
Service ID: none

##### Parameters

in	<i>Instance</i>	ADC Instance number
in	<i>Config</i>	configuration that need to apply

##### Returns

void

Definition at line 415 of file Adc\_Hal.c.

#### 4.4.2.4 Adc\_Hal\_ConfigAutoCalibration()

```
void Adc_Hal_ConfigAutoCalibration (
    uint8 Instance )
```

Config autocalibration.

##### Note

Function ID: DES\_ADC\_API\_023  
Service ID: none

##### Parameters

in	<i>Instance</i>	ADC Instance number
----	-----------------	---------------------

##### Returns

adc base address ptr

Definition at line 528 of file Adc\_Hal.c.

#### 4.4.2.5 Adc\_Hal\_ConfigChannel()

```
void Adc_Hal_ConfigChannel (
    uint8 Instance,
    const Adc_ChainConfigType * Config )
```

adc config the Channel args to the hw.

##### Note

Function ID: DES\_ADC\_API\_003  
Service ID: none

##### Parameters

in	<i>Instance</i>	ADC Instance number
in	<i>Config</i>	configuration that need to apply

##### Returns

void

Definition at line 240 of file Adc\_Hal.c.

#### 4.4.2.6 Adc\_Hal\_ConfigConverter()

```
void Adc_Hal_ConfigConverter (
    uint8 Instance,
    const Adc_ConverterConfigType * Config )
```

adc Config convert args to hw

##### Note

Function ID: DES\_ADC\_API\_002  
Service ID: none

##### Parameters

in	<i>Instance</i>	ADC Instance number
in	<i>Config</i>	convert Config that need to Config

##### Returns

void

Definition at line 202 of file Adc\_Hal.c.

#### 4.4.2.7 Adc\_Hal\_ConfigGroup()

```
void Adc_Hal_ConfigGroup (
    uint8 Instance,
    const Adc_GroupConfigType * Config )
```

: Apply the Group Mode configuration to the hardware.

##### Note

Function ID: DES\_ADC\_API\_004  
Service ID: none

##### Parameters

in	<i>Instance</i>	ADC Instance number
in	<i>Config</i>	configuration that need to apply to the group

##### Returns

: void

Definition at line 367 of file Adc\_Hal.c.

#### 4.4.2.8 Adc\_Hal\_ConfigInjectGroup()

```
void Adc_Hal_ConfigInjectGroup (
    uint32 Instance,
    Adc_GroupConfigType * Config )
```

: Apply the Inject Group Mode configuration to the hardware.

##### Note

Function ID: DES\_ADC\_API\_19  
Service ID: none

##### Parameters

in	<i>Instance</i>	ADC Instance number
in	<i>Config</i>	configuration that need to apply to the group

##### Returns

: void

Definition at line 294 of file Adc\_Hal.c.

#### 4.4.2.9 Adc\_Hal\_ConfigRegularGroup()

```
void Adc_Hal_ConfigRegularGroup (
    uint32 Instance,
    Adc_GroupConfigType * Config )
```

: Apply the Regular Group Mode configuration to the hardware.

##### Note

Function ID: DES\_ADC\_API\_18  
Service ID: none

##### Parameters

in	<i>Instance</i>	ADC Instance number
in	<i>Config</i>	configuration that need to apply to the group

##### Returns

: void

Definition at line 284 of file Adc\_Hal.c.

#### 4.4.2.10 Adc\_Hal\_ConvertToTemperature()

```
float32 Adc_Hal_ConvertToTemperature (
    const uint16 Value,
    const float RefVoltage )
```

Convert ADC value of T-Sensor to Temperature.

##### Note

Function ID: DES\_ADC\_API\_025  
Service ID: none

##### Parameters

in	<i>Value</i>	ADC conversion result of internal T-Sensor
in	<i>RefVoltage</i>	reference voltage of ADC, in mV

##### Returns

temperature value in celsius scale

Definition at line 509 of file Adc\_Hal.c.

#### 4.4.2.11 Adc\_Hal\_Deinit()

```
void Adc_Hal_Deinit (
    uint8 Instance )
```

Deinitialize ADC module hw.

##### Note

Function ID: DES\_ADC\_API\_001  
Service ID: none

##### Parameters

in	<i>Instance</i>	ADC Instance number
----	-----------------	---------------------

##### Returns

void

Definition at line 175 of file Adc\_Hal.c.

#### 4.4.2.12 Adc\_Hal\_DmaEnable()

```
void Adc_Hal_DmaEnable (
    uint32 Instance,
    boolean Enable )
```

: Enable adc-dma control.

##### Note

Function ID: DES\_ADC\_API\_20  
Service ID: none

##### Parameters

in	<i>Instance</i>	ADC Instance number
in	<i>Enable</i>	Enable flag.

##### Returns

: void

Definition at line 304 of file Adc\_Hal.c.



#### 4.4.2.13 Adc\_Hal\_GetBase()

```
ADC_Type* Adc_Hal_GetBase (
    uint8 Instance )
```

Get the adc base address.

##### Note

Function ID: DES\_ADC\_API\_022  
Service ID: none

##### Parameters

in	<i>Instance</i>	ADC Instance number
----	-----------------	---------------------

##### Returns

adc base address ptr

Definition at line 593 of file Adc\_Hal.c.

#### 4.4.2.14 Adc\_Hal\_GetConvertCompleteFlag()

```
boolean Adc_Hal_GetConvertCompleteFlag (
    uint8 Instance,
    Adc_SequenceType SeqIndex )
```

Get conversion complete flag.

##### Note

Function ID: DES\_ADC\_API\_010  
Service ID: none

##### Parameters

in	<i>Instance</i>	ADC Instance number
in	<i>SeqIndex</i>	ADC group sequence index

##### Returns

whether ADC group sequence index complete, -TRUE: ADC convert complete -FALSE: ADC convert not complete

Definition at line 440 of file Adc\_Hal.c.

#### 4.4.2.15 Adc\_Hal\_GetIdleFlag()

```
boolean Adc_Hal_GetIdleFlag (
    uint8 Instance )
```

Get ADC Idle flag.

##### Note

Function ID: DES\_ADC\_API\_009  
Service ID: none

##### Parameters

in	<i>Instance</i>	ADC Instance number
----	-----------------	---------------------

##### Returns

whether ADC is idle, -TRUE: ADC is not converting -FALSE: ADC is busy converting

Definition at line 433 of file Adc\_Hal.c.

#### 4.4.2.16 Adc\_Hal\_GetParityVal()

```
uint8 Adc_Hal_GetParityVal (
    uint8 Instance,
    Adc_SequenceType SeqIndex )
```

Get ADC group sequence index parity value.

##### Note

Function ID: DES\_ADC\_API\_21  
Service ID: none

##### Parameters

in	<i>Instance</i>	ADC Instance number
in	<i>SeqIndex</i>	ADC group sequence index

##### Returns

ADC group sequence index parity value

Definition at line 491 of file Adc\_Hal.c.

#### 4.4.2.17 Adc\_Hal\_GetSeqResult()

```
uint16 Adc_Hal_GetSeqResult (
    uint8 Instance,
    Adc_SequenceType SeqIndex )
```

Get the ADC conversion result for the group sequence.

##### Note

Function ID: DES\_ADC\_API\_008  
Service ID: none

##### Parameters

in	<i>Instance</i>	ADC Instance number
in	<i>SeqIndex</i>	ADC group sequence index

##### Returns

ADC conversion result

Definition at line 327 of file Adc\_Hal.c.

#### 4.4.2.18 Adc\_Hal\_GetTriggerConflictFlag()

```
boolean Adc_Hal_GetTriggerConflictFlag (
    uint8 Instance )
```

Get the trigger conflict flags.

##### Note

Function ID: DES\_ADC\_API\_012  
Service ID: none

##### Parameters

in	<i>Instance</i>	ADC Instance number
----	-----------------	---------------------

##### Returns

whether ADC trigger conflict, -TRUE: ADC trigger conflict -FALSE: ADC trigger not conflict

Definition at line 475 of file Adc\_Hal.c.

#### 4.4.2.19 Adc\_Hal\_Init()

```
void Adc_Hal_Init (
    uint8 Instance,
    const Adc\_InitConfigType * Config )
```

init ADC module about [Adc\\_InitConfigType](#) struct to hw

##### Note

Function ID: DES\_ADC\_API\_000  
Service ID: none

##### Parameters

in	<i>Instance</i>	ADC Instance number
in	<i>Config</i>	ADC init config args

##### Returns

void

Definition at line 163 of file Adc\_Hal.c.

#### 4.4.2.20 Adc\_Hal\_InitAMOStruct()

```
void Adc_Hal_InitAMOStruct (
    Adc\_AmoConfigType *const Config )
```

initialize Analog Monitor config structure

##### Note

Function ID: DES\_ADC\_API\_017  
Service ID: none

##### Parameters

out	<i>Config</i>	pointer to the config that need to be initialized
-----	---------------	---

##### Returns

none

Definition at line 396 of file Adc\_Hal.c.

#### 4.4.2.21 Adc\_Hal\_InitChanStruct()

```
void Adc_Hal_InitChanStruct (
    Adc_ChanConfigType *const Config )
```

Initialize adc channel config struct default args.

##### Note

Function ID: DES\_ADC\_API\_016  
Service ID: none

##### Parameters

out	<i>Config</i>	adc channel Config structure that need to be initialized
-----	---------------	--

##### Returns

none

Definition at line 229 of file Adc\_Hal.c.

#### 4.4.2.22 Adc\_Hal\_InitConverterStruct()

```
void Adc_Hal_InitConverterStruct (
    Adc_ConverterConfigType *const Config )
```

Initialize converter config struct default args.

##### Note

Function ID: DES\_ADC\_API\_014  
Service ID: none

##### Parameters

out	<i>Config</i>	converter config structure that need to be initialized
-----	---------------	--

##### Returns

none

Definition at line 188 of file Adc\_Hal.c.

#### 4.4.2.23 Adc\_Hal\_InitGroupStruct()

```
void Adc_Hal_InitGroupStruct (
    Adc_GroupConfigType *const Config )
```

Initialize adc amo config struct default args.

**Note**

Function ID: DES\_ADC\_API\_015  
Service ID: none

**Parameters**

out	<i>Config</i>	adc amo Config structure that need to be initialized
-----	---------------	--

**Returns**

none

Definition at line 347 of file Adc\_Hal.c.

**4.4.2.24 Adc\_Hal\_SetInjectOffset()**

```
void Adc_Hal_SetInjectOffset (
    uint8 Instance,
    Adc_SequenceType SeqIndex,
    uint16 Offset )
```

Set the adc inject group offset.

**Note**

Function ID: DES\_ADC\_API\_024  
Service ID: none

**Parameters**

in	<i>Instance</i>	ADC Instance number
in	<i>SeqIndex</i>	ADC group sequeunce index
in	<i>Offset</i>	ADC inject group offset value

**Returns**

void

Definition at line 519 of file Adc\_Hal.c.

**4.4.2.25 Adc\_Hal\_SwTriggerInjectConvert()**

```
void Adc_Hal_SwTriggerInjectConvert (
    uint8 Instance )
```

Start software trigger inject group conversion.

**Note**

Function ID: DES\_ADC\_API\_007  
Service ID: none

## Parameters

in	<i>Instance</i>	ADC Instance number
----	-----------------	---------------------

## Returns

void

Definition at line 319 of file Adc\_Hal.c.

## 4.4.2.26 Adc\_Hal\_SwTriggerRegularConvert()

```
void Adc_Hal_SwTriggerRegularConvert (
    uint8 Instance )
```

Start software trigger regular group conversion.

## Note

Function ID: DES\_ADC\_API\_006

Service ID: none

## Parameters

in	<i>Instance</i>	ADC Instance number
----	-----------------	---------------------

## Returns

void

Definition at line 311 of file Adc\_Hal.c.

## 4.5 Adc\_Hal\_Types.h File Reference

This file provides adc hal types header.

```
#include "Device_Register.h"
```

## Classes

- struct [Adc\\_Average\\_ConfigType](#)  
*ADC hardware average structure.*
- struct [Adc\\_InterruptInfoType](#)  
*ADC interrupt information structure.*
- struct [Adc\\_ConverterConfigType](#)  
*ADC general conversion config.*

- struct [Adc\\_ChanConfigType](#)  
*Defines the control Channel configuration.*
- struct [Adc\\_GroupConfigType](#)  
*Defines the control group configuration.*
- struct [Adc\\_AmoConfigType](#)  
*ADC analog monitor(AMO) config structure.*
- struct [Adc\\_InterruptType](#)  
*ADC interrupt struct.*
- struct [Adc\\_InitConfigType](#)  
*ADC init config args structure.*

## Macros

- #define [BIT\(x\)](#) (1UL << (uint32)(x))
- #define [ADC\\_REGULAR\\_SEQUENCE\\_LENGTH](#) (24UL)
- #define [ADC\\_INJECT\\_SEQUENCE\\_LENGTH](#) (4UL)
- #define [ADC\\_GET\\_AVERAGE\\_NUM\(x\)](#) (4UL << (x))
- #define [ADC\\_CTM\\_REGULAR\\_SOURCE\\_COUNT](#) (4UL)
- #define [ADC\\_CTM\\_INJECT\\_SOURCE\\_COUNT](#) (4UL)

## Typedefs

- typedef void(\* [Adc\\_CallbackType](#)) ([Adc\\_InterruptInfoType](#) \*Info)  
*ADC interrupt handler type.*

## Enumerations

- enum [Adc\\_PrescaleType](#) {  
[ADC\\_CLK\\_DIVIDE\\_1](#) = 0UL, [ADC\\_CLK\\_DIVIDE\\_2](#), [ADC\\_CLK\\_DIVIDE\\_3](#), [ADC\\_CLK\\_DIVIDE\\_4](#),  
[ADC\\_CLK\\_DIVIDE\\_5](#), [ADC\\_CLK\\_DIVIDE\\_6](#), [ADC\\_CLK\\_DIVIDE\\_7](#), [ADC\\_CLK\\_DIVIDE\\_8](#),  
[ADC\\_CLK\\_DIVIDE\\_9](#), [ADC\\_CLK\\_DIVIDE\\_10](#), [ADC\\_CLK\\_DIVIDE\\_11](#), [ADC\\_CLK\\_DIVIDE\\_12](#),  
[ADC\\_CLK\\_DIVIDE\\_13](#), [ADC\\_CLK\\_DIVIDE\\_14](#), [ADC\\_CLK\\_DIVIDE\\_15](#), [ADC\\_CLK\\_DIVIDE\\_16](#) }  
*Type of clock prescaler factor.*
- enum [Adc\\_ResolutionType](#) { [ADC\\_RESOLUTION\\_12BIT](#) = 0U, [ADC\\_RESOLUTION\\_10BIT](#), [ADC\\_RESOLUTION\\_8BIT](#), [ADC\\_RESOLUTION\\_MAX](#) = 3U }  
*Type of Channel Resolution in number of bits.*
- enum [Adc\\_VoltageReferenceType](#) { [ADC\\_VOLTAGEREF\\_VREF](#) = 0U, [ADC\\_VOLTAGEREF\\_VDDA](#) }  
*Voltage reference selection.*
- enum [Adc\\_SamplingTimeType](#) {  
[ADC\\_SPT\\_CLK\\_5](#) = 0U, [ADC\\_SPT\\_CLK\\_10](#), [ADC\\_SPT\\_CLK\\_15](#), [ADC\\_SPT\\_CLK\\_23](#),  
[ADC\\_SPT\\_CLK\\_35](#), [ADC\\_SPT\\_CLK\\_45](#), [ADC\\_SPT\\_CLK\\_85](#), [ADC\\_SPT\\_CLK\\_185](#) = 7U,  
[ADC\\_SPT\\_CLK\\_MAX](#) = 8U }  
*Type of sampling time, i.e. the time during which the Value is sampled, (in clock- cycles).*
- enum [Adc\\_GroupConvType](#) { [ADC\\_REGULAR\\_GROUP](#) = 0U, [ADC\\_INJECT\\_GROUP](#), [ADC\\_GROUP\\_MAX](#) }  
*ADC conversion group type enumeration.*
- enum [Adc\\_SequenceType](#) {  
[ADC\\_RSEQ\\_0](#) = 0U, [ADC\\_RSEQ\\_1](#), [ADC\\_RSEQ\\_2](#), [ADC\\_RSEQ\\_3](#),  
[ADC\\_RSEQ\\_4](#), [ADC\\_RSEQ\\_5](#), [ADC\\_RSEQ\\_6](#), [ADC\\_RSEQ\\_7](#),  
[ADC\\_RSEQ\\_8](#), [ADC\\_RSEQ\\_9](#), [ADC\\_RSEQ\\_10](#), [ADC\\_RSEQ\\_11](#),  
[ADC\\_RSEQ\\_12](#), [ADC\\_RSEQ\\_13](#), [ADC\\_RSEQ\\_14](#), [ADC\\_RSEQ\\_15](#),  
[ADC\\_RSEQ\\_16](#), [ADC\\_RSEQ\\_17](#), [ADC\\_RSEQ\\_18](#), [ADC\\_RSEQ\\_19](#),  
[ADC\\_RSEQ\\_20](#), [ADC\\_RSEQ\\_21](#), [ADC\\_RSEQ\\_22](#), [ADC\\_RSEQ\\_23](#),  
[ADC\\_RSEQ\\_24](#), [ADC\\_RSEQ\\_25](#), [ADC\\_RSEQ\\_26](#), [ADC\\_RSEQ\\_27](#),  
[ADC\\_RSEQ\\_28](#), [ADC\\_RSEQ\\_29](#), [ADC\\_RSEQ\\_30](#), [ADC\\_RSEQ\\_31](#),  
[ADC\\_ISEQ\\_0](#), [ADC\\_ISEQ\\_1](#), [ADC\\_ISEQ\\_2](#), [ADC\\_ISEQ\\_3](#),  
[ADC\\_ISEQ\\_MAX](#) }



*ADC sequence type enumeration.*

- enum [Adc\\_InputChannelType](#) {  
[ADC\\_CH\\_0](#) = 0U, [ADC\\_CH\\_1](#), [ADC\\_CH\\_2](#), [ADC\\_CH\\_3](#),  
[ADC\\_CH\\_4](#), [ADC\\_CH\\_5](#), [ADC\\_CH\\_6](#), [ADC\\_CH\\_7](#),  
[ADC\\_CH\\_8](#), [ADC\\_CH\\_9](#), [ADC\\_CH\\_10](#), [ADC\\_CH\\_11](#),  
[ADC\\_CH\\_12](#), [ADC\\_CH\\_13](#), [ADC\\_CH\\_14](#), [ADC\\_CH\\_15](#),  
[ADC\\_CH\\_16](#), [ADC\\_CH\\_17](#), [ADC\\_CH\\_18](#), [ADC\\_CH\\_19](#),  
[ADC\\_CH\\_20](#), [ADC\\_CH\\_21](#), [ADC\\_CH\\_22](#), [ADC\\_CH\\_23](#),  
[ADC\\_CH\\_24](#), [ADC\\_CH\\_25](#), [ADC\\_CH\\_26](#), [ADC\\_CH\\_27](#),  
[ADC\\_CH\\_28](#), [ADC\\_CH\\_29](#), [ADC\\_CH\\_30](#), [ADC\\_CH\\_31](#),  
[ADC\\_CH\\_MAX](#), [ADC\\_CH\\_DISABLE](#) = 0x3FU }

*ADC Channel type enumeration.*

- enum [Adc\\_AverageType](#) { [ADC\\_AVERAGE\\_4](#) = 0U, [ADC\\_AVERAGE\\_8](#) = 1U, [ADC\\_AVERAGE\\_16](#) = 2U, [ADC\\_AVERAGE\\_32](#) = 3U }

*Hardware average selection.*

- enum [Adc\\_InterleaveType](#) { [ADC\\_INTERLEAVE\\_DISABLE](#) = 0U, [ADC\\_INTERLEAVE\\_0](#) = 1U, [ADC\\_INTERLEAVE\\_1](#) = 3U }

*Hardware average selection.*

- enum [Adc\\_ResultAlignmentType](#) { [ADC\\_ALIGN\\_RIGHT](#) = 0x00U, [ADC\\_ALIGN\\_LEFT](#) }

*Type for Alignment of ADC raw results in ADC result buffer (left/right Alignment).*

- enum [Adc\\_AmoTriggerModeType](#) { [ADC\\_AMO\\_TRIGGER\\_LEVEL](#) = 0U, [ADC\\_AMO\\_TRIGGER\\_EDGE](#) }

*ADC analog monitor trigger mode enumeration.*

- enum [Adc\\_TriggerSourceType](#) { [ADC\\_TRIGG\\_SRC\\_SW](#) = 0U, [ADC\\_TRIGG\\_SRC\\_HW](#) }

*the trigger source for an ADC Channel group.*

- enum [Adc\\_EventType](#) {  
[ADC\\_EVENT\\_AMO](#) = [ADC\\_STR\\_AMO\\_Msk](#), [ADC\\_EVENT\\_AAMO](#) = [ADC\\_STR\\_AAMO\\_Msk](#), [ADC\\_EVENT\\_NAMO](#) = [ADC\\_STR\\_NAMO\\_Msk](#), [ADC\\_EVENT\\_EOC](#) = 0x80U,  
[ADC\\_EVENT\\_IEOC](#) = 0x8000U }

*Events which can trigger ADC Callback.*

## 4.5.1 Detailed Description

This file provides adc hal types header.

## 4.5.2 Macro Definition Documentation

### 4.5.2.1 ADC\_CTU\_INJECT\_SOURCE\_COUNT

```
#define ADC_CTU_INJECT_SOURCE_COUNT (4UL)
```

Definition at line 70 of file [Adc\\_Hal\\_Types.h](#).

### 4.5.2.2 ADC\_CTU\_REGULAR\_SOURCE\_COUNT

```
#define ADC_CTU_REGULAR_SOURCE_COUNT (4UL)
```

ADC injection group trigger source count number in CTU

Definition at line 68 of file [Adc\\_Hal\\_Types.h](#).

#### 4.5.2.3 ADC\_GET\_AVERAGE\_NUM

```
#define ADC_GET_AVERAGE_NUM(  
    x ) (4UL << (x))
```

ADC regular group trigger source count number in CTU

Definition at line 63 of file Adc\_Hal\_Types.h.

#### 4.5.2.4 ADC\_INJECT\_SEQUENCE\_LENGTH

```
#define ADC_INJECT_SEQUENCE_LENGTH (4UL)
```

get ADC average count number, x should be Adc\_AverageType

Definition at line 60 of file Adc\_Hal\_Types.h.

#### 4.5.2.5 ADC\_REGULAR\_SEQUENCE\_LENGTH

```
#define ADC_REGULAR_SEQUENCE_LENGTH (24UL)
```

ADC injected group sequence length.

Definition at line 57 of file Adc\_Hal\_Types.h.

#### 4.5.2.6 BIT

```
#define BIT(  
    x ) (1UL << (uint32)(x))
```

ADC regular group sequence length.

Definition at line 50 of file Adc\_Hal\_Types.h.

### 4.5.3 Typedef Documentation

#### 4.5.3.1 Adc\_CallbackType

```
typedef void(* Adc_CallbackType) (Adc_InterruptInfoType *Info)
```

ADC interrupt handler type.

Definition at line 392 of file Adc\_Hal\_Types.h.

### 4.5.4 Enumeration Type Documentation

#### 4.5.4.1 Adc\_AmoTriggerModeType

```
enum Adc_AmoTriggerModeType
```

ADC analog monitor trigger mode enumeration.

**Enumerator**

ADC_AMO_TRIGGER_LEVEL	AMO Level trigger
ADC_AMO_TRIGGER_EDGE	AMO Edge trigger

Definition at line 341 of file Adc\_Hal\_Types.h.

**4.5.4.2 Adc\_AverageType**

```
enum Adc\_AverageType
```

Hardware average selection.

**Enumerator**

ADC_AVERAGE_4	Hardware average of 4 samples.
ADC_AVERAGE_8	Hardware average of 8 samples.
ADC_AVERAGE_16	Hardware average of 16 samples.
ADC_AVERAGE_32	Hardware average of 32 samples.

Definition at line 311 of file Adc\_Hal\_Types.h.

**4.5.4.3 Adc\_EventType**

```
enum Adc\_EventType
```

Events which can trigger ADC Callback.

**Enumerator**

ADC_EVENT_AMO	Abnormal Event in AMO level mode
ADC_EVENT_AAMO	Abnormal Event in AMO edge mode
ADC_EVENT_NAMO	Normal Event in AMO edge mode
ADC_EVENT_EOC	End of conversion event
ADC_EVENT_IEOC	End of inject conversion event

Definition at line 359 of file Adc\_Hal\_Types.h.

**4.5.4.4 Adc\_GroupConvType**

```
enum Adc\_GroupConvType
```

ADC conversion group type enumeration.

## Enumerator

ADC_REGULAR_GROUP	ADC regular group
ADC_INJECT_GROUP	ADC inject group
ADC_GROUP_MAX	ADC group max Value, should be less than this Value

Definition at line 136 of file Adc\_Hal\_Types.h.

## 4.5.4.5 Adc\_InputChannelType

enum [Adc\\_InputChannelType](#)

ADC Channel type enumeration.

## Enumerator

ADC_CH_0	ADC Channel 0
ADC_CH_1	ADC Channel 1
ADC_CH_2	ADC Channel 2
ADC_CH_3	ADC Channel 3
ADC_CH_4	ADC Channel 4
ADC_CH_5	ADC Channel 5
ADC_CH_6	ADC Channel 6
ADC_CH_7	ADC Channel 7
ADC_CH_8	ADC Channel 8
ADC_CH_9	ADC Channel 9
ADC_CH_10	ADC Channel 10
ADC_CH_11	ADC Channel 11
ADC_CH_12	ADC Channel 12
ADC_CH_13	ADC Channel 13
ADC_CH_14	ADC Channel 14
ADC_CH_15	ADC Channel 15
ADC_CH_16	ADC Channel 16
ADC_CH_17	ADC Channel 17
ADC_CH_18	ADC Channel 18
ADC_CH_19	ADC Channel 19
ADC_CH_20	ADC Channel 20
ADC_CH_21	ADC Channel 21
ADC_CH_22	ADC Channel 22
ADC_CH_23	ADC Channel 23
ADC_CH_24	ADC Channel 24
ADC_CH_25	ADC Channel 25
ADC_CH_26	ADC Channel 26
ADC_CH_27	ADC Channel 27
ADC_CH_28	ADC Channel 28
ADC_CH_29	ADC Channel 29
ADC_CH_30	ADC Channel 30
ADC_CH_31	ADC Channel 31
ADC_CH_MAX	Invalid Channel
ADC_CH_DISABLE	Disabled Channel

Definition at line 190 of file Adc\_Hal\_Types.h.

#### 4.5.4.6 Adc\_InterleaveType

enum [Adc\\_InterleaveType](#)

Hardware average selection.

##### Enumerator

ADC_INTERLEAVE_DISABLE	ADC interleave disable.
ADC_INTERLEAVE_0	ADC0/1 interleave PB13/PB0.
ADC_INTERLEAVE_1	ADC0/1 interleave PB14/PB1.

Definition at line 322 of file Adc\_Hal\_Types.h.

#### 4.5.4.7 Adc\_PrescaleType

enum [Adc\\_PrescaleType](#)

Type of clock prescaler factor.

##### Enumerator

ADC_CLK_DIVIDE_1	Input clock divided by 1.
ADC_CLK_DIVIDE_2	Input clock divided by 2.
ADC_CLK_DIVIDE_3	Input clock divided by 3.
ADC_CLK_DIVIDE_4	Input clock divided by 4.
ADC_CLK_DIVIDE_5	Input clock divided by 5.
ADC_CLK_DIVIDE_6	Input clock divided by 6.
ADC_CLK_DIVIDE_7	Input clock divided by 7.
ADC_CLK_DIVIDE_8	Input clock divided by 8.
ADC_CLK_DIVIDE_9	Input clock divided by 9.
ADC_CLK_DIVIDE_10	Input clock divided by 10.
ADC_CLK_DIVIDE_11	Input clock divided by 11.
ADC_CLK_DIVIDE_12	Input clock divided by 12.
ADC_CLK_DIVIDE_13	Input clock divided by 13.
ADC_CLK_DIVIDE_14	Input clock divided by 14.
ADC_CLK_DIVIDE_15	Input clock divided by 15.
ADC_CLK_DIVIDE_16	Input clock divided by 16.

Definition at line 76 of file Adc\_Hal\_Types.h.

#### 4.5.4.8 Adc\_ResolutionType

enum [Adc\\_ResolutionType](#)

Type of Channel Resolution in number of bits.

##### Enumerator

ADC_RESOLUTION_12BIT	12-bit Resolution mode
ADC_RESOLUTION_10BIT	10-bit Resolution mode
ADC_RESOLUTION_8BIT	8-bit Resolution mode
ADC_RESOLUTION_MAX	Invalid Resolution mode

Definition at line 99 of file [Adc\\_Hal\\_Types.h](#).

#### 4.5.4.9 Adc\_ResultAlignmentType

enum [Adc\\_ResultAlignmentType](#)

Type for Alignment of ADC raw results in ADC result buffer (left/right Alignment).

##### Enumerator

ADC_ALIGN_RIGHT	AMO Level trigger
ADC_ALIGN_LEFT	AMO Level trigger

Definition at line 332 of file [Adc\\_Hal\\_Types.h](#).

#### 4.5.4.10 Adc\_SamplingTimeType

enum [Adc\\_SamplingTimeType](#)

Type of sampling time, i.e. the time during which the Value is sampled, (in clock- cycles).

##### Enumerator

ADC_SPT_CLK_5	ADC Sample time 5 Clock Cycle
ADC_SPT_CLK_10	ADC Sample time 10 Clock Cycle
ADC_SPT_CLK_15	ADC Sample time 15 Clock Cycle
ADC_SPT_CLK_23	ADC Sample time 23 Clock Cycle
ADC_SPT_CLK_35	ADC Sample time 35 Clock Cycle
ADC_SPT_CLK_45	ADC Sample time 45 Clock Cycle
ADC_SPT_CLK_85	ADC Sample time 85 Clock Cycle
ADC_SPT_CLK_185	ADC Sample time 185 Clock Cycle
ADC_SPT_CLK_MAX	Invalid ADC Sample time

Definition at line 120 of file Adc\_Hal\_Types.h.

#### 4.5.4.11 Adc\_SequenceType

enum [Adc\\_SequenceType](#)

ADC sequence type enumeration.

##### Enumerator

ADC_RSEQ_0	ADC rseq 0
ADC_RSEQ_1	ADC rseq 1
ADC_RSEQ_2	ADC rseq 2
ADC_RSEQ_3	ADC rseq 3
ADC_RSEQ_4	ADC rseq 4
ADC_RSEQ_5	ADC rseq 5
ADC_RSEQ_6	ADC rseq 6
ADC_RSEQ_7	ADC rseq 7
ADC_RSEQ_8	ADC rseq 8
ADC_RSEQ_9	ADC rseq 9
ADC_RSEQ_10	ADC rseq 10
ADC_RSEQ_11	ADC rseq 11
ADC_RSEQ_12	ADC rseq 12
ADC_RSEQ_13	ADC rseq 13
ADC_RSEQ_14	ADC rseq 14
ADC_RSEQ_15	ADC rseq 15
ADC_RSEQ_16	ADC rseq 16
ADC_RSEQ_17	ADC rseq 17
ADC_RSEQ_18	ADC rseq 18
ADC_RSEQ_19	ADC rseq 19
ADC_RSEQ_20	ADC rseq 20
ADC_RSEQ_21	ADC rseq 21
ADC_RSEQ_22	ADC rseq 22
ADC_RSEQ_23	ADC rseq 23
ADC_RSEQ_24	ADC rseq 23
ADC_RSEQ_25	ADC rseq 23
ADC_RSEQ_26	ADC rseq 23
ADC_RSEQ_27	ADC rseq 23
ADC_RSEQ_28	ADC rseq 23
ADC_RSEQ_29	ADC rseq 23
ADC_RSEQ_30	ADC rseq 23
ADC_RSEQ_31	ADC rseq 23
ADC_ISEQ_0	ADC iseq 0
ADC_ISEQ_1	ADC iseq 1
ADC_ISEQ_2	ADC iseq 2
ADC_ISEQ_3	ADC iseq 3
ADC_ISEQ_MAX	MAX Seq index

Definition at line 146 of file Adc\_Hal\_Types.h.

#### 4.5.4.12 Adc\_TriggerSourceType

enum [Adc\\_TriggerSourceType](#)

the trigger source for an ADC Channel group.

##### Enumerator

ADC_TRIGG_SRC_SW	Internal software trigger
ADC_TRIGG_SRC_HW	External signal trigger

Definition at line 350 of file Adc\_Hal\_Types.h.

#### 4.5.4.13 Adc\_VoltageReferenceType

enum [Adc\\_VoltageReferenceType](#)

Voltage reference selection.

##### Enumerator

ADC_VOLTAGEREF_VREF	VrefH and VrefL as Voltage reference.
ADC_VOLTAGEREF_VDDA	VDDA as Voltage reference.

Definition at line 110 of file Adc\_Hal\_Types.h.



# Index

AC784xx\_API\_Reference\_Manual\_ADC.pdf, [56](#)

AC784xx\_Adc\_Reg.h, [17](#)

ADC\_EACH\_REG\_SPT\_NUM, [20](#)

Adc\_Reg\_BusClkEnCtrl, [20](#)

Adc\_Reg\_ClearInjectEOCFlag, [21](#)

Adc\_Reg\_ClearRegularEOCFlag, [21](#)

Adc\_Reg\_ClearSTRFlag, [22](#)

Adc\_Reg\_GetAverageEnableFlag, [22](#)

Adc\_Reg\_GetAverageMode, [23](#)

Adc\_Reg\_GetDMAEnableFlag, [23](#)

Adc\_Reg\_GetGEOEVINEnableFlag, [25](#)

Adc\_Reg\_GetGEOEVIN, [25](#)

Adc\_Reg\_GetGainOffset0Value, [24](#)

Adc\_Reg\_GetGainOffset1Value, [24](#)

Adc\_Reg\_GetInjectData, [26](#)

Adc\_Reg\_GetInjectEOCFlag, [26](#)

Adc\_Reg\_GetInjectEOCInterruptEnableFlag, [27](#)

Adc\_Reg\_GetInjectLength, [27](#)

Adc\_Reg\_GetInjectOffset, [28](#)

Adc\_Reg\_GetInjectParityVal, [28](#)

Adc\_Reg\_GetRegularData, [30](#)

Adc\_Reg\_GetRegularEOCFlag, [30](#)

Adc\_Reg\_GetRegularEOCInterruptEnableFlag, [31](#)

Adc\_Reg\_GetRegularLength, [31](#)

Adc\_Reg\_GetRegularParityVal, [32](#)

Adc\_Reg\_GetSTRFlag, [33](#)

Adc\_Reg\_SetAMOIInjectMode, [33](#)

Adc\_Reg\_SetAMOInterrupt, [34](#)

Adc\_Reg\_SetAMOOffset, [34](#)

Adc\_Reg\_SetAMORegularMode, [35](#)

Adc\_Reg\_SetAMOSingleChannel, [35](#)

Adc\_Reg\_SetAMOSingleChannelMode, [36](#)

Adc\_Reg\_SetAMOThreshold, [36](#)

Adc\_Reg\_SetAMOTriggerMode, [37](#)

Adc\_Reg\_SetAverageEnableFlag, [37](#)

Adc\_Reg\_SetAverageMode, [38](#)

Adc\_Reg\_SetCalibrationEnableFlag, [38](#)

Adc\_Reg\_SetChannelSampleTime, [39](#)

Adc\_Reg\_SetClockPrescaler, [39](#)

Adc\_Reg\_SetContinuousEnableFlag, [40](#)

Adc\_Reg\_SetDMAEnableFlag, [42](#)

Adc\_Reg\_SetDataAlign, [41](#)

Adc\_Reg\_SetGEOEVINEnableFlag, [44](#)

Adc\_Reg\_SetGEOEVIN, [43](#)

Adc\_Reg\_SetGainOffset0Value, [42](#)

Adc\_Reg\_SetGainOffset1Value, [43](#)

Adc\_Reg\_SetInjectAutoEnableFlag, [44](#)

Adc\_Reg\_SetInjectConversionChannel, [44](#)

Adc\_Reg\_SetInjectDiscontinuousEnableFlag, [45](#)

Adc\_Reg\_SetInjectEOCInterruptEnableFlag, [46](#)

Adc\_Reg\_SetInjectLength, [46](#)

Adc\_Reg\_SetInjectOffset, [47](#)

Adc\_Reg\_SetInjectTriggerSource, [47](#)

Adc\_Reg\_SetIntervalEnableFlag, [48](#)

Adc\_Reg\_SetPowerEnableFlag, [48](#)

Adc\_Reg\_SetRegularConversionChannel, [49](#)

Adc\_Reg\_SetRegularDiscontinuousEnableFlag, [50](#)

Adc\_Reg\_SetRegularDiscontinuousNum, [50](#)

Adc\_Reg\_SetRegularEOCInterruptEnableFlag, [51](#)

Adc\_Reg\_SetRegularLength, [51](#)

Adc\_Reg\_SetRegularTriggerSource, [52](#)

Adc\_Reg\_SetResolution, [52](#)

Adc\_Reg\_SetScanEnableFlag, [53](#)

Adc\_Reg\_SetVoltageReference, [54](#)

Adc\_Reg\_SoftwareStartInjectConvert, [54](#)

Adc\_Reg\_SoftwareStartRegularConvert, [55](#)

Adc\_Reg\_SwRstClkCtrl, [55](#)

ADC\_CTU\_INJECT\_SOURCE\_COUNT

Adc\_Hal\_Types.h, [88](#)

ADC\_CTU\_REGULAR\_SOURCE\_COUNT

Adc\_Hal\_Types.h, [88](#)

ADC\_EACH\_REG\_SPT\_NUM

AC784xx\_Adc\_Reg.h, [20](#)

ADC\_GET\_AVERAGE\_NUM

Adc\_Hal\_Types.h, [88](#)

ADC\_INJECT\_SEQUENCE\_LENGTH

Adc\_Hal\_Types.h, [89](#)

ADC\_REGULAR\_SEQUENCE\_LENGTH

Adc\_Hal\_Types.h, [89](#)

Adc\_AmoConfigType, [3](#)

AmoInjectEn, [3](#)

AmoInterruptEn, [3](#)

AmoLowOffset, [4](#)

AmoLowThreshold, [4](#)

AmoRegularEn, [4](#)

AmoSingleChannel, [4](#)

AmoSingleModeEn, [4](#)

AmoTriggerMode, [5](#)

AmoUpOffset, [5](#)

AmoUpThreshold, [5](#)

Adc\_AmoTriggerModeType

Adc\_Hal\_Types.h, [89](#)

Adc\_Average\_ConfigType, [5](#)

Enable, [6](#)

Value, [6](#)

Adc\_AverageType

Adc\_Hal\_Types.h, [90](#)

Adc\_CallbackType

Adc\_Hal\_Types.h, [89](#)

Adc\_ChanConfigType, [7](#)

Channel, [8](#)

InternalChannelSrc, [8](#)

InterruptEn, [8](#)

SeqIndex, [8](#)

- Spt, 8
- Adc\_ConverterConfigType, 9
  - Alignment, 9
  - ClockDivide, 9
  - DmaEnable, 9
  - HwAverage, 10
  - Interleave, 10
  - PowerEn, 10
  - Resolution, 10
  - VoltageRef, 10
- Adc\_EventType
  - Adc\_Hal\_Types.h, 90
- Adc\_GroupConfigType, 11
  - ContinuousModeEn, 11
  - DmaArgs, 12
  - DmaCallback, 12
  - DmaDstAddr, 12
  - InjectAutoModeEn, 12
  - InjectDiscontinuousModeEn, 12
  - InjectSequenceLength, 13
  - InjectTrigger, 13
  - IntervalModeEn, 13
  - RegularDiscontinuousModeEn, 13
  - RegularDiscontinuousNum, 13
  - RegularSequenceLength, 14
  - RegularTrigger, 14
  - ScanModeEn, 14
- Adc\_GroupConvType
  - Adc\_Hal\_Types.h, 90
- Adc\_Hal.c, 56
  - Adc\_Hal\_ClearConvertCompleteFlag, 59
  - Adc\_Hal\_ClearTriggerConflictFlag, 59
  - Adc\_Hal\_ConfigAmo, 60
  - Adc\_Hal\_ConfigAutoCalibration, 60
  - Adc\_Hal\_ConfigChannel, 61
  - Adc\_Hal\_ConfigConverter, 61
  - Adc\_Hal\_ConfigGroup, 62
  - Adc\_Hal\_ConfigInjectGroup, 62
  - Adc\_Hal\_ConfigRegularGroup, 63
  - Adc\_Hal\_ConvertToTemperature, 63
  - Adc\_Hal\_Deinit, 64
  - Adc\_Hal\_DmaEnable, 64
  - Adc\_Hal\_GetBase, 65
  - Adc\_Hal\_GetConvertCompleteFlag, 65
  - Adc\_Hal\_GetIdleFlag, 66
  - Adc\_Hal\_GetParityVal, 66
  - Adc\_Hal\_GetSeqResult, 67
  - Adc\_Hal\_GetTriggerConflictFlag, 67
  - Adc\_Hal\_Init, 68
  - Adc\_Hal\_InitAMOStruct, 68
  - Adc\_Hal\_InitChanStruct, 69
  - Adc\_Hal\_InitConverterStruct, 69
  - Adc\_Hal\_InitGroupStruct, 70
  - Adc\_Hal\_SetInjectOffset, 70
  - Adc\_Hal\_SwTriggerInjectConvert, 71
  - Adc\_Hal\_SwTriggerRegularConvert, 71
  - Ana\_GeoecalVinType, 58
  - ISR, 71, 72
  - ROUND, 58
- Adc\_Hal.h, 72
  - Adc\_Hal\_ClearConvertCompleteFlag, 73
  - Adc\_Hal\_ClearTriggerConflictFlag, 74
  - Adc\_Hal\_ConfigAmo, 74
  - Adc\_Hal\_ConfigAutoCalibration, 75
  - Adc\_Hal\_ConfigChannel, 75
  - Adc\_Hal\_ConfigConverter, 76
  - Adc\_Hal\_ConfigGroup, 76
  - Adc\_Hal\_ConfigInjectGroup, 77
  - Adc\_Hal\_ConfigRegularGroup, 77
  - Adc\_Hal\_ConvertToTemperature, 78
  - Adc\_Hal\_Deinit, 78
  - Adc\_Hal\_DmaEnable, 79
  - Adc\_Hal\_GetBase, 79
  - Adc\_Hal\_GetConvertCompleteFlag, 80
  - Adc\_Hal\_GetIdleFlag, 80
  - Adc\_Hal\_GetParityVal, 81
  - Adc\_Hal\_GetSeqResult, 81
  - Adc\_Hal\_GetTriggerConflictFlag, 82
  - Adc\_Hal\_Init, 82
  - Adc\_Hal\_InitAMOStruct, 83
  - Adc\_Hal\_InitChanStruct, 83
  - Adc\_Hal\_InitConverterStruct, 84
  - Adc\_Hal\_InitGroupStruct, 84
  - Adc\_Hal\_SetInjectOffset, 85
  - Adc\_Hal\_SwTriggerInjectConvert, 85
  - Adc\_Hal\_SwTriggerRegularConvert, 86
- Adc\_Hal\_ClearConvertCompleteFlag
  - Adc\_Hal.c, 59
  - Adc\_Hal.h, 73
- Adc\_Hal\_ClearTriggerConflictFlag
  - Adc\_Hal.c, 59
  - Adc\_Hal.h, 74
- Adc\_Hal\_ConfigAmo
  - Adc\_Hal.c, 60
  - Adc\_Hal.h, 74
- Adc\_Hal\_ConfigAutoCalibration
  - Adc\_Hal.c, 60
  - Adc\_Hal.h, 75
- Adc\_Hal\_ConfigChannel
  - Adc\_Hal.c, 61
  - Adc\_Hal.h, 75
- Adc\_Hal\_ConfigConverter
  - Adc\_Hal.c, 61
  - Adc\_Hal.h, 76
- Adc\_Hal\_ConfigGroup
  - Adc\_Hal.c, 62
  - Adc\_Hal.h, 76
- Adc\_Hal\_ConfigInjectGroup
  - Adc\_Hal.c, 62
  - Adc\_Hal.h, 77
- Adc\_Hal\_ConfigRegularGroup
  - Adc\_Hal.c, 63
  - Adc\_Hal.h, 77
- Adc\_Hal\_ConvertToTemperature
  - Adc\_Hal.c, 63
  - Adc\_Hal.h, 78
- Adc\_Hal\_Deinit
  - Adc\_Hal.c, 64
  - Adc\_Hal.h, 78
- Adc\_Hal\_DmaEnable
  - Adc\_Hal.c, 64
  - Adc\_Hal.h, 79

Adc\_Hal\_GetBase  
     Adc\_Hal.c, 65  
     Adc\_Hal.h, 79  
 Adc\_Hal\_GetConvertCompleteFlag  
     Adc\_Hal.c, 65  
     Adc\_Hal.h, 80  
 Adc\_Hal\_GetIdleFlag  
     Adc\_Hal.c, 66  
     Adc\_Hal.h, 80  
 Adc\_Hal\_GetParityVal  
     Adc\_Hal.c, 66  
     Adc\_Hal.h, 81  
 Adc\_Hal\_GetSeqResult  
     Adc\_Hal.c, 67  
     Adc\_Hal.h, 81  
 Adc\_Hal\_GetTriggerConflictFlag  
     Adc\_Hal.c, 67  
     Adc\_Hal.h, 82  
 Adc\_Hal\_Init  
     Adc\_Hal.c, 68  
     Adc\_Hal.h, 82  
 Adc\_Hal\_InitAMOStruct  
     Adc\_Hal.c, 68  
     Adc\_Hal.h, 83  
 Adc\_Hal\_InitChanStruct  
     Adc\_Hal.c, 69  
     Adc\_Hal.h, 83  
 Adc\_Hal\_InitConverterStruct  
     Adc\_Hal.c, 69  
     Adc\_Hal.h, 84  
 Adc\_Hal\_InitGroupStruct  
     Adc\_Hal.c, 70  
     Adc\_Hal.h, 84  
 Adc\_Hal\_SetInjectOffset  
     Adc\_Hal.c, 70  
     Adc\_Hal.h, 85  
 Adc\_Hal\_SwTriggerInjectConvert  
     Adc\_Hal.c, 71  
     Adc\_Hal.h, 85  
 Adc\_Hal\_SwTriggerRegularConvert  
     Adc\_Hal.c, 71  
     Adc\_Hal.h, 86  
 Adc\_Hal\_Types.h, 86  
     ADC\_CTU\_INJECT\_SOURCE\_COUNT, 88  
     ADC\_CTU\_REGULAR\_SOURCE\_COUNT, 88  
     ADC\_GET\_AVERAGE\_NUM, 88  
     ADC\_INJECT\_SEQUENCE\_LENGTH, 89  
     ADC\_REGULAR\_SEQUENCE\_LENGTH, 89  
     Adc\_AmoTriggerModeType, 89  
     Adc\_AverageType, 90  
     Adc\_CallbackType, 89  
     Adc\_EventType, 90  
     Adc\_GroupConvType, 90  
     Adc\_InputChannelType, 91  
     Adc\_InterleaveType, 92  
     Adc\_PrescaleType, 92  
     Adc\_ResolutionType, 92  
     Adc\_ResultAlignmentType, 93  
     Adc\_SamplingTimeType, 93  
     Adc\_SequenceType, 94  
     Adc\_TriggerSourceType, 95  
     Adc\_VoltageReferenceType, 95  
     BIT, 89  
 Adc\_InitConfigType, 14  
     Interrupt, 15  
 Adc\_InputChannelType  
     Adc\_Hal\_Types.h, 91  
 Adc\_InterleaveType  
     Adc\_Hal\_Types.h, 92  
 Adc\_InterruptInfoType, 15  
     Event, 15  
     Instance, 16  
     sequence, 16  
 Adc\_InterruptType, 16  
     Callback, 16  
 Adc\_PrescaleType  
     Adc\_Hal\_Types.h, 92  
 Adc\_Reg\_BusClkEnCtrl  
     AC784xx\_Adc\_Reg.h, 20  
 Adc\_Reg\_ClearInjectEOCFlag  
     AC784xx\_Adc\_Reg.h, 21  
 Adc\_Reg\_ClearRegularEOCFlag  
     AC784xx\_Adc\_Reg.h, 21  
 Adc\_Reg\_ClearSTRFlag  
     AC784xx\_Adc\_Reg.h, 22  
 Adc\_Reg\_GetAverageEnableFlag  
     AC784xx\_Adc\_Reg.h, 22  
 Adc\_Reg\_GetAverageMode  
     AC784xx\_Adc\_Reg.h, 23  
 Adc\_Reg\_GetDMAEnableFlag  
     AC784xx\_Adc\_Reg.h, 23  
 Adc\_Reg\_GetGEOEVINEnableFlag  
     AC784xx\_Adc\_Reg.h, 25  
 Adc\_Reg\_GetGEOEVIN  
     AC784xx\_Adc\_Reg.h, 25  
 Adc\_Reg\_GetGainOffset0Value  
     AC784xx\_Adc\_Reg.h, 24  
 Adc\_Reg\_GetGainOffset1Value  
     AC784xx\_Adc\_Reg.h, 24  
 Adc\_Reg\_GetInjectData  
     AC784xx\_Adc\_Reg.h, 26  
 Adc\_Reg\_GetInjectEOCFlag  
     AC784xx\_Adc\_Reg.h, 26  
 Adc\_Reg\_GetInjectEOCInterruptEnableFlag  
     AC784xx\_Adc\_Reg.h, 27  
 Adc\_Reg\_GetInjectLength  
     AC784xx\_Adc\_Reg.h, 27  
 Adc\_Reg\_GetInjectOffset  
     AC784xx\_Adc\_Reg.h, 28  
 Adc\_Reg\_GetInjectParityVal  
     AC784xx\_Adc\_Reg.h, 28  
 Adc\_Reg\_GetRegularData  
     AC784xx\_Adc\_Reg.h, 30  
 Adc\_Reg\_GetRegularEOCFlag  
     AC784xx\_Adc\_Reg.h, 30  
 Adc\_Reg\_GetRegularEOCInterruptEnableFlag  
     AC784xx\_Adc\_Reg.h, 31  
 Adc\_Reg\_GetRegularLength  
     AC784xx\_Adc\_Reg.h, 31  
 Adc\_Reg\_GetRegularParityVal  
     AC784xx\_Adc\_Reg.h, 32  
 Adc\_Reg\_GetSTRFlag

AC784xx\_Adc\_Reg.h, [33](#)  
 Adc\_Reg\_SetAMOIInjectMode  
     AC784xx\_Adc\_Reg.h, [33](#)  
 Adc\_Reg\_SetAMOIInterrupt  
     AC784xx\_Adc\_Reg.h, [34](#)  
 Adc\_Reg\_SetAMOOOffset  
     AC784xx\_Adc\_Reg.h, [34](#)  
 Adc\_Reg\_SetAMORegularMode  
     AC784xx\_Adc\_Reg.h, [35](#)  
 Adc\_Reg\_SetAMOSingleChannel  
     AC784xx\_Adc\_Reg.h, [35](#)  
 Adc\_Reg\_SetAMOSingleChannelMode  
     AC784xx\_Adc\_Reg.h, [36](#)  
 Adc\_Reg\_SetAMOThreshold  
     AC784xx\_Adc\_Reg.h, [36](#)  
 Adc\_Reg\_SetAMOTriggerMode  
     AC784xx\_Adc\_Reg.h, [37](#)  
 Adc\_Reg\_SetAverageEnableFlag  
     AC784xx\_Adc\_Reg.h, [37](#)  
 Adc\_Reg\_SetAverageMode  
     AC784xx\_Adc\_Reg.h, [38](#)  
 Adc\_Reg\_SetCalibrationEnableFlag  
     AC784xx\_Adc\_Reg.h, [38](#)  
 Adc\_Reg\_SetChannelSampleTime  
     AC784xx\_Adc\_Reg.h, [39](#)  
 Adc\_Reg\_SetClockPrescaler  
     AC784xx\_Adc\_Reg.h, [39](#)  
 Adc\_Reg\_SetContinuousEnableFlag  
     AC784xx\_Adc\_Reg.h, [40](#)  
 Adc\_Reg\_SetDMAEnableFlag  
     AC784xx\_Adc\_Reg.h, [42](#)  
 Adc\_Reg\_SetDataAlign  
     AC784xx\_Adc\_Reg.h, [41](#)  
 Adc\_Reg\_SetGEOEVINEnableFlag  
     AC784xx\_Adc\_Reg.h, [44](#)  
 Adc\_Reg\_SetGEOEVIN  
     AC784xx\_Adc\_Reg.h, [43](#)  
 Adc\_Reg\_SetGainOffset0Value  
     AC784xx\_Adc\_Reg.h, [42](#)  
 Adc\_Reg\_SetGainOffset1Value  
     AC784xx\_Adc\_Reg.h, [43](#)  
 Adc\_Reg\_SetInjectAutoEnableFlag  
     AC784xx\_Adc\_Reg.h, [44](#)  
 Adc\_Reg\_SetInjectConversionChannel  
     AC784xx\_Adc\_Reg.h, [44](#)  
 Adc\_Reg\_SetInjectDiscontinuousEnableFlag  
     AC784xx\_Adc\_Reg.h, [45](#)  
 Adc\_Reg\_SetInjectEOCInterruptEnableFlag  
     AC784xx\_Adc\_Reg.h, [46](#)  
 Adc\_Reg\_SetInjectLength  
     AC784xx\_Adc\_Reg.h, [46](#)  
 Adc\_Reg\_SetInjectOffset  
     AC784xx\_Adc\_Reg.h, [47](#)  
 Adc\_Reg\_SetInjectTriggerSource  
     AC784xx\_Adc\_Reg.h, [47](#)  
 Adc\_Reg\_SetIntervalEnableFlag  
     AC784xx\_Adc\_Reg.h, [48](#)  
 Adc\_Reg\_SetPowerEnableFlag  
     AC784xx\_Adc\_Reg.h, [48](#)  
 Adc\_Reg\_SetRegularConversionChannel  
     AC784xx\_Adc\_Reg.h, [49](#)  
 Adc\_Reg\_SetRegularDiscontinuousEnableFlag  
     AC784xx\_Adc\_Reg.h, [50](#)  
 Adc\_Reg\_SetRegularDiscontinuousNum  
     AC784xx\_Adc\_Reg.h, [50](#)  
 Adc\_Reg\_SetRegularEOCInterruptEnableFlag  
     AC784xx\_Adc\_Reg.h, [51](#)  
 Adc\_Reg\_SetRegularLength  
     AC784xx\_Adc\_Reg.h, [51](#)  
 Adc\_Reg\_SetRegularTriggerSource  
     AC784xx\_Adc\_Reg.h, [52](#)  
 Adc\_Reg\_SetResolution  
     AC784xx\_Adc\_Reg.h, [52](#)  
 Adc\_Reg\_SetScanEnableFlag  
     AC784xx\_Adc\_Reg.h, [53](#)  
 Adc\_Reg\_SetVoltageReference  
     AC784xx\_Adc\_Reg.h, [54](#)  
 Adc\_Reg\_SoftwareStartInjectConvert  
     AC784xx\_Adc\_Reg.h, [54](#)  
 Adc\_Reg\_SoftwareStartRegularConvert  
     AC784xx\_Adc\_Reg.h, [55](#)  
 Adc\_Reg\_SwRstClkCtrl  
     AC784xx\_Adc\_Reg.h, [55](#)  
 Adc\_ResolutionType  
     Adc\_Hal\_Types.h, [92](#)  
 Adc\_ResultAlignmentType  
     Adc\_Hal\_Types.h, [93](#)  
 Adc\_SamplingTimeType  
     Adc\_Hal\_Types.h, [93](#)  
 Adc\_SequenceType  
     Adc\_Hal\_Types.h, [94](#)  
 Adc\_TriggerSourceType  
     Adc\_Hal\_Types.h, [95](#)  
 Adc\_VoltageReferenceType  
     Adc\_Hal\_Types.h, [95](#)  
 adc\_calibration\_t, [6](#)  
     userGain, [7](#)  
     userOffset, [7](#)  
 Alignment  
     Adc\_ConverterConfigType, [9](#)  
 AmoInjectEn  
     Adc\_AmoConfigType, [3](#)  
 AmoInterruptEn  
     Adc\_AmoConfigType, [3](#)  
 AmoLowOffset  
     Adc\_AmoConfigType, [4](#)  
 AmoLowThreshold  
     Adc\_AmoConfigType, [4](#)  
 AmoRegularEn  
     Adc\_AmoConfigType, [4](#)  
 AmoSingleChannel  
     Adc\_AmoConfigType, [4](#)  
 AmoSingleModeEn  
     Adc\_AmoConfigType, [4](#)  
 AmoTriggerMode  
     Adc\_AmoConfigType, [5](#)  
 AmoUpOffset  
     Adc\_AmoConfigType, [5](#)  
 AmoUpThreshold  
     Adc\_AmoConfigType, [5](#)  
 Ana\_GeoecalVinType  
     Adc\_Hal.c, [58](#)

BIT  
    Adc\_Hal\_Types.h, [89](#)

Callback  
    Adc\_InterruptType, [16](#)

Channel  
    Adc\_ChanConfigType, [8](#)

ClockDivide  
    Adc\_ConverterConfigType, [9](#)

ContinuousModeEn  
    Adc\_GroupConfigType, [11](#)

DmaArgs  
    Adc\_GroupConfigType, [12](#)

DmaCallback  
    Adc\_GroupConfigType, [12](#)

DmaDstAddr  
    Adc\_GroupConfigType, [12](#)

DmaEnable  
    Adc\_ConverterConfigType, [9](#)

Enable  
    Adc\_Average\_ConfigType, [6](#)

Event  
    Adc\_InterruptInfoType, [15](#)

HwAverage  
    Adc\_ConverterConfigType, [10](#)

ISR  
    Adc\_Hal.c, [71](#), [72](#)

InjectAutoModeEn  
    Adc\_GroupConfigType, [12](#)

InjectDiscontinuousModeEn  
    Adc\_GroupConfigType, [12](#)

InjectSequenceLength  
    Adc\_GroupConfigType, [13](#)

InjectTrigger  
    Adc\_GroupConfigType, [13](#)

Instance  
    Adc\_InterruptInfoType, [16](#)

Interleave  
    Adc\_ConverterConfigType, [10](#)

InternalChannelSrc  
    Adc\_ChanConfigType, [8](#)

Interrupt  
    Adc\_InitConfigType, [15](#)

InterruptEn  
    Adc\_ChanConfigType, [8](#)

IntervalModeEn  
    Adc\_GroupConfigType, [13](#)

PowerEn  
    Adc\_ConverterConfigType, [10](#)

ROUND  
    Adc\_Hal.c, [58](#)

RegularDiscontinuousModeEn  
    Adc\_GroupConfigType, [13](#)

RegularDiscontinuousNum  
    Adc\_GroupConfigType, [13](#)

RegularSequenceLength  
    Adc\_GroupConfigType, [14](#)

RegularTrigger  
    Adc\_GroupConfigType, [14](#)

Resolution  
    Adc\_ConverterConfigType, [10](#)

ScanModeEn  
    Adc\_GroupConfigType, [14](#)

SeqIndex  
    Adc\_ChanConfigType, [8](#)

sequence  
    Adc\_InterruptInfoType, [16](#)

Spt  
    Adc\_ChanConfigType, [8](#)

userGain  
    adc\_calibration\_t, [7](#)

userOffset  
    adc\_calibration\_t, [7](#)

Value  
    Adc\_Average\_ConfigType, [6](#)

VoltageRef  
    Adc\_ConverterConfigType, [10](#)