ADC Driver
User Guide
V1.00.02

Support Chips:

ISD9160

Support Platforms:

Nuvoton

Content

1. ADC Driver

1.1 ADC Introduction

The ISD91XX series includes a 2nd Order Delta-Sigma Audio Analog-to-Digital converter providing SNR >85dB and THD >70dB. The converter can run at sampling rates up to 6.144MHz while a configurable decimation filter allows over sampling ratios of 64/128/192 and 384. This provides support for standard audio sampling rates from 8kHz to 48kHz.

1.2 ADC Feature

- Front-end PGA providing gain range of -12dB – 51dB.
- Configurable OSR (Over Sampling Ratio) of 64/128/192/384
- Configurable clock rate through master oscillator integer division.
- Decimation signal can be used directly or passed to biquad filter for further filtering.
- Audio data buffered to 8 word FIFO, accessible via APB and PDMA.

1.3 Type Definition

E_DRVADC_PDHIRES

Enumeration Identifier	Value	Description
eDRVADC_PDHIRES_CONNECTED	0	Connect the high resistance reference to VMID.
eDRVADC_PDHIRES_DISCONNECTED	1	The high resistance reference is disconnected from VMID.

E_DRVADC_PDLORES

Enumeration Identifier	Value	Description
eDRVADC_PDLORES_CONNECTED	0	Connect the low resistance reference to VMID.
eDRVADC_PDLORES_DISCONNECTED	1	The low resistance reference is disconnected from VMID.

E_DRVADC_PULLDOWN

Enumeration Identifier	Value	Description
eDRVADC_PULLDOWN_VMID_RELEASE	0	Release VMID pin for reference operation.
eDRVADC_PULLDOWN_VMID_GND	1	Pull VMID pin to ground.

E_DRVADC_BOOSTGAIN

Enumeration Identifier	Value	Description
eDRVADC_BOOSTGAIN_0DB	0	Boost amplifier gain is 0 dB.
eDRVADC_BOOSTGAIN_26DB	1	Boost amplifier gain is 26 dB.

E_DRVADC_PU_BOOST

Enumeration Identifier	Value	Description
eDRVADC_PU_BOOST_OFF	0	Boost amplifier is powered off.
eDRVADC_PU_BOOST_ON	1	Boost amplifier is powered on.

E_DRVADC_PU_PGA

Enumeration Identifier	Value	Description
eDRVADC_PU_PGA_OFF	0	PGA is powered off.

eDRVADC_PU_PGA_ON	1	PGA is powered on.

E_DRVADC_REF_SEL

Enumeration Identifier	Value	Description
eDRVADC_REF_SEL_VMID	0	ADC reference voltage is VMID.
eDRVADC_REF_SEL_VBG	1	ADC reference voltage is VBG.

E_DRVADC_PU_IBGEN

Enumeration Identifier	Value	Description
eDRVADC_PU_IBGEN_OFF	0	IBG is powered off.
eDRVADC_PU_IBGEN_ON	1	IBG is powered on.

E_DRVADC_PU_MOD

Enumeration Identifier	Value	Description
eDRVADC_PU_MOD_OFF	0	ADC modulator is powered off.
eDRVADC_PU_MOD_ON	1	ADC modulator is powered on.

E_DRVADC_PU_BUFADC

Enumeration Identifier	Value	Description
eDRVADC_PU_BUFADC_OFF	0	ADC buffer is powered off.
eDRVADC_PU_BUFADC_ON	1	ADC buffer is powered on.

E_DRVADC_PU_BUFPGA

Enumeration Identifier	Value	Description
eDRVADC_PU_BUFPGA_OFF	0	PGA buffer is powered off.
eDRVADC_PU_BUFPGA_ON	1	PGA buffer is powered on.

E_DRVADC_PU_ZCD

Enumeration Identifier	Value	Description
eDRVADC_PU_ZCD_OFF	0	Zero cross detect comparator is powered off.
eDRVADC_PU_ZCD_ON	1	Zero cross detect comparator is powered on.

E_DRVADC_MIC_BIAS_SEL

Enumeration Identifier	Value	Description
eDRVADC_MIC_BIAS_90_VCCA	0	90% of VCCA

eDRVADC_MIC_BIAS_65_VCCA	1	65% of VCCA
eDRVADC_MIC_BIAS_75_VCCA	2	75% of VCCA
eDRVADC_MIC_BIAS_50_VCCA	3	50% of VCCA
eDRVADC_MIC_BIAS_2p4V	4	2.4V
eDRVADC_MIC_BIAS_1p7V	5	1.7V
eDRVADC_MIC_BIAS_2p0V	6	2.0V
eDRVADC_MIC_BIAS_1p3V	7	1.3V

E_DRVADC_MUXP_SEL

Enumeration Identifier	Value	Description
eDRVADC_MUXP_NONE	0	PGA postive input select none.
eDRVADC_MUXP_GPB1_SEL	1	PGA postive input select GPB1.
eDRVADC_MUXP_GPB3_SEL	2	PGA postive input select GPB3
eDRVADC_MUXP_GPB5_SEL	4	PGA postive input select GPB5
eDRVADC_MUXP_GPB7_SEL	8	PGA postive input select GPB7.

E_DRVADC_MUXN_SEL

Enumeration Identifier	Value	Description
eDRVADC_MUXN_NONE	0	PGA negative input select none.
eDRVADC_MUXN_GPB0_SEL	0x01	PGA negative input select GPB0.
eDRVADC_MUXN_GPB1_SEL	0x02	PGA negative input select GPB1.
eDRVADC_MUXN_GPB2_SEL	0x04	PGA negative input select GPB2.
eDRVADC_MUXN_GPB3_SEL	0x08	PGA negative input select GPB3
eDRVADC_MUXN_GPB4_SEL	0x10	PGA negative input select GPB4.
eDRVADC_MUXN_GPB5_SEL	0x20	PGA negative input select GPB5
eDRVADC_MUXN_GPB6_SEL	0x40	PGA negative input select GPB6.
eDRVADC_MUXN_GPB7_SEL	0x80	PGA negative input select GPB7.

E_DRVADC_INPUT_MODE

Enumeration Identifier	Value	Description
eDRVADC_SINGLE_END_CH0_IN_N	0	Select ch0 in negative PGA input
eDRVADC_SINGLE_END_CH1_IN_N	1	Select ch1 in negative PGA input
eDRVADC_SINGLE_END_CH1_IN_P	2	Select ch1 in positive PGA input
eDRVADC_SINGLE_END_CH2_IN_N	3	Select ch2 in negative PGA input
eDRVADC_SINGLE_END_CH3_IN_N	4	Select ch3 in negative PGA input
eDRVADC_SINGLE_END_CH3_IN_P	5	Select ch3 in positive PGA input

eDRVADC_SINGLE_END_CH4_IN_N	6	Select ch4 in negative PGA input
eDRVADC_SINGLE_END_CH5_IN_N	7	Select ch5 in negative PGA input
eDRVADC_SINGLE_END_CH5_IN_P	8	Select ch5 in positive PGA input
eDRVADC_SINGLE_END_CH6_IN_N	9	Select ch6 in negative PGA input
eDRVADC_SINGLE_END_CH7_IN_N	10	Select ch7 in negative PGA input
eDRVADC_SINGLE_END_CH7_IN_P	11	Select ch7 in positive PGA input
eDRVADC_DIFFERENTIAL_CH01	12	Select ch0 in negative PGA input and ch1 in positive PGA input.
eDRVADC_DIFFERENTIAL_CH23	13	Select ch2 in negative PGA input and ch3 in positive PGA input.
eDRVADC_DIFFERENTIAL_CH45	14	Select ch4 in negative PGA input and ch5 in positive PGA input.
eDRVADC_DIFFERENTIAL_CH67	15	Select ch6 in negative PGA input and ch7 in positive PGA input.
eDRVADC_DIFFERENTIAL	16	Reserved for MIC or Temp input selection.

E_DRVADC_INPUT_SRC

Enumeration Identifier	Value	Description
eDRVADC_MIC	0	PGA input select MIC.
eDRVADC_GPIO	1	PGA input select GPIOB.
eDRVADC_TEMP	2	PGA input select temperature sense.

E_DRVADC_OSR

Enumeration Identifier	Value	Description
eDRVADC_OSR_64	0	ADC over sampling ratio is 64.
eDRVADC_OSR_128	1	ADC over sampling ratio is 128.
eDRVADC_OSR_192	2	ADC over sampling ratio is 192.
eDRVADC_OSR_384	3	ADC over sampling ratio is 384.

E_DRVADC_COMP_CONDITION

Enumeration Identifier	Value	Description
eDRVADC_LESS_THAN	0	Less than the compare data.
eDRVADC_GREATER_OR_EQUAL	1	Greater or equal to the compare data.

E_DRVADC_BIQ_SELPATH

Enumeration Identifier	Value	Description

eDRVADC_BIQ_IN_ADC	0	Biquad filter is in ADC path.
eDRVADC_BIQ_IN_DPWM	1	Biquad filter is in DPWM path.

Error Code Identifier	Value	Description
E_DRVADC_ARGUMENT	1	Invalid argument
E_DRVADC_CHANNELNUM	2	Non-support channel number
E_DRVADC_OPEN_BEFORE	3	ADC has been opened before.
E_DRVADC_ANA_OPEN	10	Analog block is not opened.

1.4 Functions

DrvADC_AnaOpen

Prototype

ERRCODE

DrvADC_AnaOpen(void);

Description

Open analog block clock for config.

Parameters

None

Include

Driver\DrvADC.h

Return Value

E_SUCCESS: Success

Example

DrvADC_AnaOpen()

DrvADC_AnaClose

Prototype

void

DrvADC_AnaClose(void);

Description

Close analog block clock and return to reset state.

Parameters

None

Include

Driver\DrvADC.h

Return Value

None

Example

DrvADC_AnaClose();

DrvADC_SetAMUX

Prototype

void

DrvADC_SetAMUX(

uint32_t u32AMUXSel,

E_DRVADC_MUXP_SEL eMUXPSel,

E_DRVADC_MUXN_SEL eMUXNSel

);

Description

Specify analog mux input source.

Parameters

u32AMUXSel [in] : specify analog input source

AMUX_MIC : select microphone input

AMUX_TEMP: select temperature input

AMUX_GPIO : select GPIO input, eMUXPSel and eMUXNSel are valid when this is selected.

AMUX_OFF : none is select for analog input.

eMUXPSel [in] : specify internal PGA positive terminal input

eDRVADC_MUXP_GPB7_SEL

eDRVADC_MUXP_GPB5_SEL

eDRVADC_MUXP_GPB3_SEL

eDRVADC_MUXP_GPB1_SEL

eMUXNSel [in] : specify internal PGA negative terminal input

eDRVADC_MUXN_GPB7_SEL

eDRVADC_MUXN_GPB6_SEL

eDRVADC_MUXN_GPB5_SEL

eDRVADC_MUXN_GPB4_SEL

eDRVADC_MUXN_GPB3_SEL

eDRVADC_MUXN_GPB2_SEL

eDRVADC_MUXN_GPB1_SEL

eDRVADC_MUXN_GPB0_SEL

Include

Driver\DrvADC.h

Return Value

None

Example

```
/* Select microphone as analog input source */
DrvADC_SetAMUX(AMUX_MIC, eDRVADC_MUXP_NONE, eDRVADC_MUXN_NONE);

/* Select temperature sensor as analog input source */
DrvADC_SetAMUX(AMUX_TEMP, eDRVADC_MUXP_NONE, eDRVADC_MUXN_NONE);

/* Select GPIO as analog input source */
DrvADC_SetAMUX(AMUX_GPIO, eDRVADC_MUXP_NONE, eDRVADC_MUXN_GPB0_SEL);
```

DrvADC_SetVMID

Prototype

```
void
DrvADC_SetVMID(
    E_DRVADC_PULLDOWN ePULLDOWN,
    E_DRVADC_PDLORES    ePDLORES,
    E_DRVADC_PDHIRES    ePDHIRES
);
```

Description

Set the pull high or pull low resistor reference for VMID.

Parameters

ePULLDOWN [in]

eDRVADC_PULLDOWN_VMID_RELEASE : Release VMID pin for reference operation.

eDRVADC_PULLDOWN_VMID_GND: Pull VMID pin to ground.

ePDLORES [in]

eDRVADC_PDLORES_CONNECTED: Connect the low resistance reference to VMID. Use this setting for fast power up of VMID . Can be turned off after 50ms to save power.

eDRVADC_PDLORES_DISCONNECTED: The low resistance reference is disconnected from VMID.

ePDHIRES [in]

eDRVADC_PDHIRES_CONNECTED: Connect the high resistance reference to VMID. Use this setting for minimum power consumption.

eDRVADC_PDHIRES_DISCONNECTED: The high resistance reference is disconnected from VMID.

Include

Driver\DrvADC.h

Return Value

None

Example

```
/* MIC circuit configuration */
DrvADC_SetVMID(
    eDRVADC_PULLDOWN_VMID_RELEASE,
    eDRVADC_PDLORES_CONNECTED,
    eDRVADC_PDHIRES_DISCONNECTED);
```

DrvADC_SetMIC

Prototype

```
void
DrvADC_SetMIC(
BOOL bEnable,
E_DRVADC_MIC_BIAS_SEL eMIC_BIAS_SEL
);
```

Description

Enable/disable the microphone and set the bias voltage.

Parameters

bEnable [in]

TRUE: enable the microphone

FALSE: disable the microphone

eMIC_BIAS_SEL [in]

Specify the bias voltage. It could be eDRVADC_MIC_BIAS_90_VCCA, eDRVADC_MIC_BIAS_65_VCCA, eDRVADC_MIC_BIAS_75_VCCA, eDRVADC_MIC_BIAS_50_VCCA, eDRVADC_MIC_BIAS_2p4V, eDRVADC_MIC_BIAS_1p7V, eDRVADC_MIC_BIAS_2p0V, eDRVADC_MIC_BIAS_1p3V

Include

Driver\DrvADC.h

Return Value

None

Example

```
/* Enable the microphone and specify the bias voltage */
DrvADC_SetMIC(TRUE, eDRVADC_MIC_BIAS_90_VCCA);
```

DrvADC_SetPGA

Prototype

```
void
DrvADC_SetPGA(
    E_DRVADC_REF_SEL     eREF_SEL,
    E_DRVADC_PU_PGA      ePU_PGA,
    E_DRVADC_PU_BOOST    ePU_BOOST,
    E_DRVADC_BOOSTGAIN eBOOSTGAIN
);
```

Description

Set boost gain, boost stage/PGA power status and Vref of PGA.

Parameters

eREF_SEL [in]

eDRVADC_REF_SEL_VMID: Select VMID voltage as analog ground reference.

eDRVADC_REF_SEL_VBG: Select Bandgap voltage as analog ground reference.

ePU_PGA [in]

eDRVADC_PU_PGA_OFF: power down

eDRVADC_PU_PGA_ON: power up

ePU_BOOST [in]

eDRVADC_PU_BOOST_OFF: power down

eDRVADC_PU_BOOST_ON: power up

eBOOSTGAIN [in]

eDRVADC_BOOSTGAIN_0DB: gain = 0 dB

eDRVADC_BOOSTGAIN_26DB: gain = 26 dB

Include

Driver\DrvADC.h

Return Value

None

Example

```
DrvADC_SetPGA(
    eDRVADC_REF_SEL_VMID,
    eDRVADC_PU_PGA_ON,
    eDRVADC_PU_BOOST_ON,
    eDRVADC_BOOSTGAIN_0DB);
```

DrvADC_SetPGAGaindB

Prototype

int32_t
DrvADC_SetPGAGaindB(
int32_t i32PGAGainIndB
);

Description

Set PGA gain in dB.

Parameters

i32PGAGainIndB [in]

i32PGAGainIndB/100 = u32PGAGain * 0.75 – 12

For Example

i32PGAGainIndB	mapping dB
3225	32.25 dB
100	1 dB
0	0 dB
-1200	-12 dB

Include

Driver\DrvADC.h

Return Value

PGA gain in dB after scale 100.

Example

```
/* Set PGA gain: -6 dB*/
DrvADC_SetPGAGaindB(-600);
```

DrvADC_PGAMute

Prototype

void
DrvADC_PGAMute(
E_DRVADC_MUTE ePGAStage
);

Description

Mute the PGA pre-amplify or boost stage.

Parameters

ePGAStage [in]

eDRVADC_MUTE_PGA : mute the pre-amplify stage

eDRVADC_MUTE_IPBOOST : mute the boost stage

Include

Driver\DrvADC.h

Return Value

None

Example

```
/* Mute PGA */
DrvADC_PGAMute(eDRVADC_MUTE_PGA);
```

DrvADC_PGAUnMute

Prototype

```
void
DrvADC_PGAUnMute(
uint32_t u32PGAStage
);
```

Description

Unmute the PGA pre-amplify or boost stage.

Parameters

ePGAStage [in]

eDRVADC_MUTE_PGA : mute the pre-amplify stage

eDRVADC_MUTE_IPBOOST : mute the boost stage

Include

Driver\DrvADC.h

Return Value

None

Example

```
/* Unmute PGA */
DrvADC_PGAUnMute(eDRVADC_MUTE_PGA);
```

DrvADC_SetPower

Prototype

```
void
DrvADC_SetPower(
    E_DRVADC_PU_MOD ePU_MOD,
    E_DRVADC_PU_IBGEN ePU_IBGEN,
    E_DRVADC_PU_BUFADC ePU_BUFADC,
    E_DRVADC_PU_BUFPGA ePU_PGA,
    E_DRVADC_PU_ZCD ePU_ZCD
);
```

Description

Specify ZCD, PGA Buffer, ADC Buffer, ADC MOD, IBG power up status.

Parameters

ePU_MOD [in] : Specify ADC sigma-delta modulator power status.

ePU_MOD_ON: power up

ePU_MOD_OFF: power down

ePU_IBGEN [in] : Specify current bias generation power status.

ePU_IBGEN_ON: power up

ePU_IBGEN_OFF: power down

ePU_BUFADC [in] : Specify ADC reference buffer power status.

ePU_BUFADC_ON: power up

ePU_BUFADC_OFF: power down

ePU_PGA [in] : Specify PGA reference buffer power status.

ePU_BUFPGA_ON: power up

ePU_BUFPGA_OFF: power down

ePU_ZCD [in] : Specify Zero Cross Detect Comparator power status.

ePU_ZCD_ON: power up

ePU_ZCD_OFF: power down

Include

Driver\DrvADC.h

Return Value

None

Example

```
/* Power control */
DrvADC_SetPower(
    eDRVADC_PU_MOD_ON,
    eDRVADC_PU_IBGEN_ON,
    eDRVADC_PU_BUFADC_ON,
    eDRVADC_PU_BUFPGA_ON,
    eDRVADC_PU_ZCD_ON);
```

DrvADC_Open

Prototype

```
ERRCODE
DrvADC_Open(
S_DRVADC_PARAM *sParam
);
```

Description

Enable the ADC function. Configure the corresponding pins of the specified channels as analog input pins.

Parameters

sParam [in]: pointer to structure for config the ADC function. It contains the following parameter.

u8AdcDivisor: Determine the ADC clock frequency.

u8SDAdcDivisor: Specify oversampling clock divisor.

eOSR: Specify SDADC over sampling ratio.

eDRVADC_OSR_64/ eDRVADC_OSR_128/ eDRVADC_OSR_192/ eDRVADC_OSR_384

u8ADCFifoIntLevel: Specify the FIFO interrupt level.

eInputSrc: Specify the analog input source.

eDRVADC_MIC/ eDRVADC_TEMP/ eDRVADC_GPIO

eInputMode: Specify the analog input mode and channel. It is used only when eInputSrc == eDRVADC_GPIO.The input mode contains:

eDRVADC_SINGLE_END_CH0_IN_N

eDRVADC_SINGLE_END_CH1_IN_N

eDRVADC_SINGLE_END_CH1_IN_P

eDRVADC_SINGLE_END_CH2_IN_N

eDRVADC_SINGLE_END_CH3_IN_N

eDRVADC_SINGLE_END_CH3_IN_P

eDRVADC_SINGLE_END_CH4_IN_N

eDRVADC_SINGLE_END_CH5_IN_N

eDRVADC_SINGLE_END_CH5_IN_P

eDRVADC_SINGLE_END_CH6_IN_N

eDRVADC_SINGLE_END_CH7_IN_N

eDRVADC_SINGLE_END_CH7_IN_P

eDRVADC_DIFFERENTIAL_CH01

eDRVADC_DIFFERENTIAL_CH23

eDRVADC_DIFFERENTIAL_CH45

eDRVADC_DIFFERENTIAL_CH67

eDRVADC_DIFFERENTIAL

Include

Driver\DrvADC.h

Return Value

E_DRVADC_ARGUMENT : Argument is invalid.

E_SUCCESS: ADC has been successfully opened.

Example

```
/* Open ADC block */
S_DRVADC_PARAM sParam;
sParam.u8AdcDivisor      = 0;
sParam.u8SDAdcDivisor = 24;
sParam.eOSR              = eDRVADC_OSR_128;
sParam.eInputSrc         = eDRVADC_MIC;
sParam.eInputMode        = eDRVADC_DIFFERENTIAL;
sParam.u8ADCFifoIntLevel = 7;
u32AdcStatus=DrvADC_Open(&sParam);
```

DrvADC_Close

Prototype

```
void
DrvADC_Close(void);
```

Description

Disable the ADC function, ADC engine clock and ADC interrupt.

Parameters

None

Include

Driver\DrvADC.h

Return Value

None

Example

```
/* Close ADC */
DrvADC_Close();
```

DrvADC_SetAdcChannel

Prototype

```
ERRCODE
DrvADC_SetAdcChannel(
E_DRVADC_INPUT_SRC eInputSrc,
E_DRVADC_INPUT_MODE eInputMode
);
```

Description

Select ADC input channels.

Parameters

eInputSrc [in]

eDRVADC_MIC/ eDRVADC_TEMP/ eDRVADC_GPIO

eInputMode [in]

eDRVADC_SINGLE_END_CH0_IN_N

eDRVADC_SINGLE_END_CH1_IN_N

eDRVADC_SINGLE_END_CH1_IN_P

eDRVADC_SINGLE_END_CH2_IN_N

eDRVADC_SINGLE_END_CH3_IN_N

eDRVADC_SINGLE_END_CH3_IN_P

eDRVADC_SINGLE_END_CH4_IN_N

eDRVADC_SINGLE_END_CH5_IN_N

eDRVADC_SINGLE_END_CH5_IN_P

eDRVADC_SINGLE_END_CH6_IN_N

eDRVADC_SINGLE_END_CH7_IN_N

eDRVADC_SINGLE_END_CH7_IN_P

eDRVADC_DIFFERENTIAL_CH01

eDRVADC_DIFFERENTIAL_CH23

eDRVADC_DIFFERENTIAL_CH45

eDRVADC_DIFFERENTIAL_CH67

eDRVADC_DIFFERENTIAL

Include

Driver\DrvADC.h

Return Value

E_DRVADC_ANA_OPEN : analog block should open before call this function.

E_DRVADC_ARGUMENT : analog input mode not support.

E_SUCCESS : set adc channel successfully.

Example

```
/* Select microphone input */
DrvADC_SetAdcChannel(eDRVADC_MIC, eDRVADC_DIFFERENTIAL);
```

DrvADC_GetConversionRate

Prototype

```
uint32_t
DrvADC_GetConversionRate(void);
```

Description

Return the A/D conversion rate (sample/second.)

Parameters

None

Include

Driver\DrvADC.h

Return Value

Conversion rate.

Example

```
/* display current ADC conversion rate */
printf("ADC conversion rate = %d\n", DrvADC_GetConversionRate());
```

DrvADC_GetConversionData

Prototype

```
uint32_t
DrvADC_GetConversionData(void);
```

Description

Get the conversion result of SDADC.

Parameters

None

Include

Driver\DrvADC.h

Return Value

16-bit conversion result.

Example

```
/* display current ADC conversion data */
printf("ADC conversion data = %d\n", DrvADC_GetConversionData());
```

DrvADC_SetAdcDivisor

Prototype

```
void
DrvADC_SetAdcDivisor(
uint8_t u8AdcDivisor
);
```

Description

Set the divisor value of ADC clock.

Parameters

u8AdcDivisor [in]

Specify the divisor value. ADC clock frequency = ADC clock source frequency / (u8AdcDivisor + 1)

Include

Driver\DrvADC.h

Return Value

None

Example

```
/* ADC clock frequency    = ADC clock source frequency / (0 + 1 )*/
DrvADC_SetAdcDivisor(0)
```

DrvADC_SetAdcOverSamplingClockDivisor

Prototype

```
void
DrvADC_SetAdcOverSamplingClockDivisor(
uint8_t u8SDAdcDivisor
);
```

Description

Set the divisor value of SDADC clock.

Parameters

u8SDAdcDivisor [in]

Specify the divisor value. SDADC clock frequency = ADC clock source frequency / (u8SDAdcDivisor)

Include

Driver\DrvADC.h

Return Value

None

Example

```
/* Set the divisor value of SDADC clock. */
DrvADC_SetAdcOverSamplingClockDivisor(24);
```

DrvADC_SetOverSamplingRatio

Prototype

```
void
DrvADC_SetOverSamplingRatio(
E_DRVADC_OSR eOSR
);
```

Description

Set the over sampling ratio of SDADC

Parameters

eOSR [in]

eDRVADC_OSR_64

eDRVADC_OSR_128

eDRVADC_OSR_192

eDRVADC_OSR_384

Include

Driver\DrvADC.h

Return Value

None

Example

```
/* Set OSR=128 */
DrvADC_SetOverSamplingRatio(eDRVADC_OSR_128);
```

DrvADC_SetCICGain

Prototype

```
void
DrvADC_SetCICGain(
uint8_t u8CICGain
);
```

Description

Set the additional digital gain from the decimation filter. An additional gain is applied to signal of u8CICGain/2

Parameters

u8CICGain [in]

Specify additional CIC gain.

Include

Driver\DrvADC.h

Return Value

None.

Example

```
DrvADC_SetCICGain(2);
```

DrvADC_SetFIFOIntLevel

Prototype

```
void
DrvADC_SetFIFOIntLevel(
uint8_t u8ADCFifoIntLevel
);
```

Description

Set the interrupt level of SDADC FIFO.

Parameters

u8ADCFifoIntLevel [in]

The ADC Fifo interrupt level

Include

Driver\DrvADC.h

Return Value

None

Example

```
/* Specify ADC FIFO interrupt level = 4 */
DrvADC_SetFIFOIntLevel(4);
```

DrvADC_EnableAdcInt

Prototype

```
void
DrvADC_EnableAdcInt(
DRVADC_ADC_CALLBACK Callback,
uint32_t u32UserData
);
```

Description

Enable the ADC interrupt and setup callback function.

Parameters

Callback [in]

The callback function of ADC interrupt.

u32UserData [in]

The user's data to pass to the callback function.

Include

Driver\DrvADC.h

Return Value

None

Example

```
/* Enable the ADC interrupt and setup callback function to AdcIntCallback . */
DrvADC_EnableAdcInt(AdcIntCallback, 0);
```

DrvADC_DisableAdcInt

Prototype

```
void
DrvADC_DisableAdcInt(void);
```

Description

Disable the ADC interrupt.

Parameters

None

Include

Driver\DrvADC.h

Return Value

None

Example

DrvADC_DisableAdcInt();

DrvADC_PdmaEnable

Prototype

void

DrvADC_PdmaEnable(void);

Description

Enable the PDMA data transfer function. When PDMA transfer is enabled, the IE bit must be set to '0'.

Parameters

None

Include

Driver\DrvADC.h

Return Value

None

Example

/* Enable the PDMA data transfer function. */

DrvADC_PdmaEnable();

DrvADC_PdmaDisable

Prototype

void

DrvADC_PdmaDisable(void);

Description

Disable the PDMA data transfer function.

Parameters

None

Include

Driver\DrvADC.h

Return Value

None

Example

```
/* Disable the PDMA data transfer function. */
DrvADC_PdmaDisable();
```

DrvADC_Adcmp0Enable

Prototype

```
ERRCODE
DrvADC_Adcmp0Enable(
E_DRVADC_COMP_CONDITION eCmpCondition,
uint16_t u16CmpData,
uint8_t u8CmpMatchCount
);
```

Description

Enable the ADC result monitor 0 and configure the compare data and match count.

Parameters

eCmpCondition [in]

eDRVADC_LESS_THAN for the condition of "less than the compare data"

eDRVADC_GREATER_OR_EQUAL for the condition of "greater or equal to the compare data."

u16CmpData [in]

Specify the compare data.

u8CmpMatchCount [in]

Specify the compare match count.

Include

Driver\DrvADC.h

Return Value

E_DRVADC_ARGUMENT: one of the input arguments is out of the range.

E_SUCCESS: the compare function is enabled.

Example

```
// Enable ADC compare 0. Compare condition: conversion result < 0x800.
DrvADC_Adcmp0Enable(eDRVADC_LESS_THAN, 0x800, u8CmpMatchCount);
```

DrvADC_Adcmp0Disable

Prototype

```
void
DrvADC_Adcmp0Disable(void);
```

Description

Disable the ADC result monitor 0.

Parameters

None

Include

Driver\DrvADC.h

Return Value

None

Example

DrvADC_Adcmp0Disable();

DrvADC_EnableAdcmp0Int

Prototype

void

DrvADC_EnableAdcmp0Int(

DRVADC_ADCMP0_CALLBACK Callback,

uint32_t u32UserData

);

Description

Enable the ADC compare 0 interrupt and setup callback function.

Parameters

Callback [in]

The callback function of ADC result monitor 0 (compare 0) interrupt.

u32UserData [in]

The user's data to pass to the callback function.

Include

Driver\DrvADC.h

Return Value

None

Example

// enable ADC compare 0 interrupt and set the callback function

DrvADC_EnableAdcmp0Int(Cmp0IntCallback, 0);

DrvADC_DisableAdcmp0Int

Prototype

void

DrvADC_DisableAdcmp0Int(void);

Description

Disable the ADC compare 0 interrupt.

Parameters

None

Include

Driver\DrvADC.h

Return Value

None

Example

DrvADC_DisableAdcmp0Int()

DrvADC_Adcmp1Enable

Prototype

ERRCODE

DrvADC_Adcmp1Enable(

E_DRVADC_COMP_CONDITION eCmpCondition,

uint16_t u16CmpData,

uint8_t u8CmpMatchCount

);

Description

Enable the ADC result monitor 1 and configure the compare data and match count.

Parameters

eCmpCondition [in]

eDRVADC_LESS_THAN for the condition of "less than the compare data"

eDRVADC_GREATER_OR_EQUAL for the condition of "greater or equal to the compare data."

u16CmpData [in]

Specify the compare data.

u8CmpMatchCount [in]

Specify the compare match count.

Include

Driver\DrvADC.h

Return Value

E_DRVADC_ARGUMENT: one of the input arguments is out of the range.

E_SUCCESS: the compare function is enabled.

Example

```
// Enable ADC compare 1. Compare condition: conversion result >= 0x800.
DrvADC_Adcmp1Enable(eDRVADC_GREATER_OR_EQUAL, 0x800,    u8CmpMatchCount);
```

DrvADC_Adcmp1Disable

Prototype

void

DrvADC_Adcmp1Disable(void);

Description

Disable the ADC result monitor 1.

Parameters

None

Include

Driver\DrvADC.h

Return Value

None

Example

DrvADC_Adcmp1Disable();

DrvADC_EnableAdcmp1Int

Prototype

void

DrvADC_EnableAdcmp1Int(

DRVADC_ADCMP1_CALLBACK Callback,

uint32_t u32UserData

);

Description

Enable the ADC compare 1 interrupt and setup callback function.

Parameters

Callback [in]

The callback function of ADC result monitor 1 (compare 1) interrupt.

u32UserData [in]

The user's data to pass to the callback function.

Include

Driver\DrvADC.h

Return Value

None

Example

// enable ADC compare 1 interrupt and set the callback function

DrvADC_EnableAdcmp1Int(Cmp1IntCallback, 0);

DrvADC_DisableAdcmp1Int

Prototype

void

DrvADC_DisableAdcmp1Int(void);

Description

Disable the ADC compare 1 interrupt.

Parameters

None

Include

Driver\DrvADC.h

Return Value

none

Example

DrvADC_DisableAdcmp1Int();

DrvADC_StartConvert

Prototype

void

DrvADC_StartConvert(void);

Description

ADC Conversion enabled.

Parameters

None

Include

Driver\DrvADC.h

Return Value

None

Example

DrvADC_StartConvert();

DrvADC_StopConvert

Prototype

void

DrvADC_StopConvert(void);

Description

Conversion stopped and ADC is reset including FIFO pointers.

Parameters

None

Include

Driver\DrvADC.h

Return Value

None

Example

DrvADC_StopConvert();

DrvADC_SetBIQ

Prototype

void

DrvADC_SetBIQ(

uint16_t u16SR_DIV,

uint8_t u8PWM_UPSR,

E_DRVADC_BIQ_SELPATH eBiqSelPath,

uint32_t u32BiqCoeff[15]

);

Description

Specify the BIQ operating sampling rate, path and config the BIQ coefficients.

Parameters

u16SR_DIV [in]

This is used to program the operating sampling rate of the biq filter. The sampling rate is defined as 49.152M/(u16SR_DIV+1).

u8PWM_UPSR [in]

This is only used when SELPATH is set to 1. The operating sample rate for the biq filter will be u8PWM_UPSR*49.152M/(u16SR_DIV+1).

eBiqSelPath [in]

AC path selection for BIQ

eDRVADC_BIQ_IN_ADC : biq filter is in ADC path.

eDRVADC_BIQ_IN_DPWM : biq filter is in DPWM path.

u32BiqCoeff[in]

BIQ coefficient array for programming the 3 stage BIQ filter. There are 5 coefficeints in each stage. Totally 15 coefficients in 3.16 format.

Include

Driver\DrvADC.h

Return Value

None

Example

DrvADC_SetBIQ(u16SR_DIV, u8PWM_UPSR, eDRVADC_BIQ_IN_ADC, u32BiqCoeff[0]);

DrvADC_BIQEnable

Prototype

void

DrvADC_BIQEnable(void);

Description

Enable the BIQ filter

Parameters

None

Include

Driver\DrvADC.h

Return Value

None

Example

DrvADC_BIQEnable();

DrvADC_BIQDisable

Prototype

void

DrvADC_BIQDisable(void);

Description

Disable the BIQ filter

Parameters

None

Include

Driver\DrvADC.h

Return Value

None

Example

DrvADC_BIQDisable();

DrvADC_RstBIQ

Prototype

void

DrvADC_RstBIQ(

BOOL bFlag

);

Description

Determine the BIQ is in reset state or active state.

Parameters

bFlag [in] : Move BIQ out of reset state

0: BIQ filter is in reset state.

1: the default coefficients will be downloaded to the coefficient ram automatically in 32 internal system clock. Processor must delay enough time before changing the coefficients or turn the BIQ on.

Include

Driver\DrvADC.h

Return Value

None

Example

/* BIQ in active state */

DrvADC_RstBIQ(1);

DrvADC_GetVersion

Prototype

uint32_t

DrvADC_GetVersion(void);

Description

Return the current version number of driver.

Parameters

None

Include

Driver\DrvADC.h

Return Value

Version number:

31:24	23:16	15:8	7:0
00000000	MAJOR_NUM	MINOR_NUM	BUILD_NUM

Example

printf("Driver version:%x\n", DrvADC_GetVersion());

2. Revision History

Version	Date	Description
1.00.01	Mar. 2011	Preliminary ADC Driver User Guide of ISD9160
1.00.02	7.July 2011	Update MIC Bias voltage option.