DPWM Driver User Guide V1.00.01

Support Chips:

ISD9160

Support Platforms:

Nuvoton

Content

Content........ 2

1. DPWM Driver........ 3

1.1 DPWM Introuction........ 3

1.2 DPWM Feature........ 3

1.3 Type Definition........ 4

1.4 Functions........ 5

DrvDPWM_Open........ 5

DrvDPWM_Close........ 5

DrvDPWM_AltPins........ 5

DrvDPWM_SetSampleRate........ 6

DrvDPWM_GetSampleRate........ 7

DrvDPWM_SetDPWMClk........ 7

DrvDPWM_SetFrequency........ 7

DrvDPWM_GetFrequency........ 8

DrvDPWM_SetDeadtime........ 8

DrvDPWM_GetDeadtime........ 9

DrvDPWM_Dither........ 9

DrvDPWM_Enable........ 10

DrvDPWM_Disable........ 10

DrvDPWM_IsFIFOFull........ 11

DrvDPWM_IsFIFOEmpty........ 11

DrvDPWM_EnablePDMA........ 12

DrvDPWM_DisablePDMA........ 12

DrvDPWM_WriteFIFO........ 13

DrvDPWM_GetVersion........ 13

2. Revision History........ 15

1. DPWM Driver

1.1 DPWM Introuction

The ISD91XX series includes a differential Class D (PWM) speaker driver capable of delivering 1W into an 8Ω load at 5V supply voltage. The driver works by up-sampling and modulating a PCM input to differentially drive the SPK+ and SPK- pins. The speaker driver operates from its own independent supply VCCSPK and VSSSPK. This supply should be well decoupled as peak currents from speaker driver are large.

1.2 DPWM Feature

- Differential Bridge-Tied-Load structure to directly drive 8Ω Speaker.
- Power delivery up to 1W @5V into 8Ω.
- Power efficiency of up to 85%.
- Configurable input sample rate.
- 16 Sample FIFO for audio output.
- PDMA data channel for streaming of PCM audio data.

1.3 Type Definition

E_DRVDPWM_FREQ

Enumeration Identifier	Value	Description
eDRVDPWM_FREQ_0	0	Carrier frequency type 0
eDRVDPWM_FREQ_1	1	Carrier frequency type 1
eDRVDPWM_FREQ_2	2	Carrier frequency type 2
eDRVDPWM_FREQ_3	3	Carrier frequency type 3
eDRVDPWM_FREQ_4	4	Carrier frequency type 4
eDRVDPWM_FREQ_5	5	Carrier frequency type 5
eDRVDPWM_FREQ_6	6	Carrier frequency type 6
eDRVDPWM_FREQ_7	7	Carrier frequency type 7

E_DRVDPWM_DEADTIME

Enumeration Identifier	Value	Description
eDRVDPWM_DEADTIME_OFF	0	Dead time is off.
eDRVDPWM_DEADTIME_ON	1	Dead time is on.

E_DRVDPWM_DPWMCLK

Enumeration Identifier	Value	Description
E_DRVDPWM_DPWMCLK_HCLK,	0	DPWM operation clock is 1xHCLK
E_DRVDPWM_DPWMCLK_HCLKX2	1	DPWM operation clock is 2xHCLK.

E_DRVDPWM_ DITHERTYPE

Enumeration Identifier	Value	Description
eDRVDPWM_DITHER_NONE	0	Dither is off.
eDRVDPWM_DITHER_ONE	1	Dither level is 1.
eDRVDPWM_DITHER_TWO	3	Dither level is 2.

Error Code Identifier	Value	Description
E_DRVDPWM_ARGUMENT	1	Invalid argument

1.4 Functions

DrvDPWM_Open

Prototype

void

DrvDPWM_Open(void);

Description

Open DPWM IP and config the parameter.

Parameters

None

Include

Driver\DrvDPWM.h

Return Value

None

Example

/* Open DPWM IP, enable the clock for operation */

DrvDPWM_Open()

DrvDPWM_Close

Prototype

void

DrvDPWM_Close(void);

Description

Close DPWM IP, return to default state and stop clock.

Parameters

None

Include

Driver\DrvDPWM.h

Return Value

None

Example

/* Close DPWM IP */

DrvDPWM_Close();

DrvDPWM_AltPins

Prototype

```
void
DrvDPWM_AltPins(
int32_t Enable
);
```

Description

Specify SPK alternate output on GPA12/13 or not.

Parameters

Enable [in]

TRUE: SPK alternate output on GPA12/13.

FALSE: GPA12/13 is normal GPIO state.

Include

Driver\DrvDPWM.h

Return Value

None

Example

```
/* Let SPK alternate output on GPA12/13 */
DrvDPWM_AltPins(TRUE);
```

DrvDPWM_SetSampleRate

Prototype

```
uint32_t
DrvDPWM_SetSampleRate(
uint32_t u32SampleRate
);
```

Description

Set the sample rate.

Parameters

u32SampleRate [in]

Sample Rate in Hz

Include

Driver\DrvDPWM.h

Return Value

Actual sample rate set.

Example

```
/* Set DPWM sample rate 16000 Hz */
DrvDPWM_SetSampleRate(16000);
```

DrvDPWM_GetSampleRate

Prototype

uint32_t

DrvDPWM_GetSampleRate(void);

Description

Get the sample rate of DPWM.

Parameters

None

Include

Driver\DrvDPWM.h

Return Value

Current sample rate DPWM is set to in Hz.

Example

/* Get the sample rate of DPWM */

printf(“Current sample rate of DPWM =%d Hz\n”, DrvDPWM_GetSampleRate());

DrvDPWM_SetDPWMClk

Prototype

void

DrvDPWM_SetDPWMClk(

E_DRVDPWM_DPWMCLK eDpwmClk

);

Description

Set DPWM clock is one or two times of system clock.

Parameters

eDpwmClk [in]

E_DRVDPWM_DPWMCLK_HCLK

E_DRVDPWM_DPWMCLK_HCLKX2

Include

Driver\DrvDPWM.h

Return Value

None

Example

/* Set DPWM clock is two times of system clock. */

DrvDPWM_SetDPWMClk(E_DRVDPWM_DPWMCLK_HCLKX2)

DrvDPWM_SetFrequency

Prototype

void

DrvDPWM_SetFrequency(

E_DRVDPWM_FREQ eFrequency

);

Description

Specify DPWM frequency setting.

Parameters

Frequency [in]

eDRVDPWM_FREQ_0~7

Include

Driver\DrvDPWM.h

Return Value

None

Example

/* Specify DPWM frequency setting */

DrvDPWM_SetFrequency(eDRVDPWM_FREQ_3);

DrvDPWM_GetFrequency

Prototype

uint32_t

DrvDPWM_GetFrequency(void);

Description

Get current DPWM frequency setting.

Parameters

None

Include

Driver\DrvDPWM.h

Return Value

Return the DPWM frequency setting.

Example

/* Get current DPWM frequency setting. */

printf("Current DPWM frequency setting=%d\n", DrvDPWM_GetFrequency());

DrvDPWM_SetDeadtime

Prototype

void

DrvDPWM_SetDeadtime(

E_DRVDPWM_DEADTIME eDeadTime

);

Description

Specify DPWM Dead time on/off.

Parameters

eDeadTime [in]

eDRVDPWM_DEADTIME_ON

eDRVDPWM_DEADTIME_OFF

Include

Driver\DrvDPWM.h

Return Value

None

Example

/* Turn DPWM dead time on */

DrvDPWM_SetDeadtime(eDRVDPWM_DEADTIME_ON);

DrvDPWM_GetDeadtime

Prototype

uint32_t

DrvDPWM_GetDeadtime(void);

Description

Get current DPWM dead time setting.

Parameters

None

Include

Driver\DrvDPWM.h

Return Value

The DPWM Dead Time setting.

Example

printf(“DPWM dead time=%d\n”, DrvDPWM_GetDeadtime());

DrvDPWM_Dither

Prototype

void

DrvDPWM_Dither(

E_DRVDPWM_DITHERTYPE eDitherType

);

Description

Specify dither type.

Parameters

eDitherType [in]

Specify the dither type. It could be

eDRVDPWM_DITHER_NONE

eDRVDPWM_DITHER_ONE

eDRVDPWM_DITHER_TWO

Include

Driver\DrvDPWM.h

Return Value

None

Example

```
/* specify dither type 1 */
DrvDPWM_Dither(eDRVDPWM_DITHER_ONE);
```

DrvDPWM_Enable

Prototype

```
void
DrvDPWM_Enable(void);
```

Description

Enable DPWM function.

Parameters

None

Include

Driver\DrvDPWM.h

Return Value

None

Example

```
/* Enable DPWM function */
DrvDPWM_Enable();
```

DrvDPWM_Disable

Prototype

```
void
DrvDPWM_Disable(void);
```

Description

Disable DPWM function.

Parameters

None

Include

Driver\DrvDPWM.h

Return Value

None

Example

```
/* Disable DPWM function */
DrvDPWM_Disable();
```

DrvDPWM_IsFIFOFull

Prototype

```
BOOL
DrvDPWM_IsFIFOFull(void);
```

Description

Check if DPWM FIFO is full or not.

Parameters

None

Include

Driver\DrvDPWM.h

Return Value

TRUE: FIFO is full.

FALSE: FIFO is not full.

Example

```
/* check DPWM FIFO status */
DrvDPWM_IsFIFOFull();
```

DrvDPWM_IsFIFOEmpty

Prototype

```
BOOL
DrvDPWM_IsFIFOEmpty(void);
```

Description

Check if DPWM FIFO is empty or not.

Parameters

None

Include

Driver\DrvDPWM.h

Return Value

TRUE: FIFO is empty.

FALSE: FIFO is not empty.

Example

/* check DPWM FIFO status */

DrvDPWM_IsFIFOEmpty();

DrvDPWM_EnablePDMA

Prototype

void

DrvDPWM_EnablePDMA(void)

Description

Enable DPWM for PDMA transfer.

Parameters

None

Include

Driver\DrvDPWM.h

Return Value

None

Example

/* Enable DPWM for PDMA transfer. */

DrvDPWM_EnablePDMA();

DrvDPWM_DisablePDMA

Prototype

void

DrvDPWM_DisablePDMA(void);

Description

Disable DPWM for PDMA transfer.

Parameters

None

Include

Driver\DrvDPWM.h

Return Value

None

Example

/* Disable DPWM for PDMA transfer. */

DrvDPWM_DisablePDMA();

DrvDPWM_WriteFIFO

Prototype

int32_t

DrvDPWM_WriteFIFO(

int32_t *pi32Stream,

int32_t i32count

);

Description

Write to DPWM FIFO for transmit.

Parameters

pi32Stream [in]

Pointer of input data stream for transmit.

i32count [in]

Transmit sample count.

Include

Driver\DrvDPWM.h

Return Value

Remain sample count not fill in fifo.

Example

int32_t Count;

Count=DrvDPWM_WriteFIFO(&SrcArray[0],16);

DrvDPWM_GetVersion

Prototype

uint32_t

DrvDPWM_GetVersion(void);

Description

Get the version number of ISD9160 DPWM driver.

Parameters

None

Include

Driver\DrvDPWM.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”, DrvDPWM_GetVersion());
```

2. Revision History

Version	Date	Description
1.00.01	Mar. 2011	Preliminary DPWM Driver User Guide of ISD9160