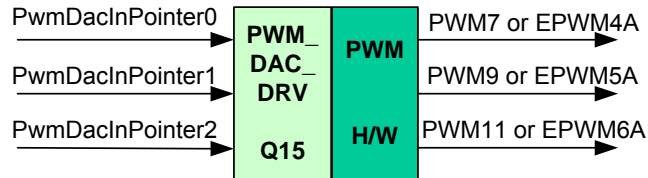


Description

This module converts any s/w variables with Q15 representation into the PWM signals in EVB for 281x or EPWM4-6 for 280x. Thus, it can be used to view the signal, represented by the variable, at the outputs of the PWM7, PWM9, PWM11 (for 281x) or EPWM4A, EPWM5A, EPWM6A (for 280x) pins through the external low-pass filters.

**Availability**

This 16-bit module is available in one interface format:

- 1) The C interface version

Module Properties

Type: Target Dependent, Application Independent

Target Devices: x281x or x280x

C Version File Names: f281xpwm dac.c, f281xpwm dac.h (for x281x)
f280xpwm dac.c, f280xpwm dac.h (for x280x)

IQmath library files for C: N/A

Item	C version	Comments
Code Size [□] (x281x/x280x)	84/264 words	
Data RAM	0 words [*]	
xDAIS ready	No	
XDAIS component	No	IALG layer not implemented
Multiple instances	Yes	
Reentrancy	Yes	

^{*} Each pre-initialized PWMDAC structure consumes 11 words in the data memory

[□] Code size mentioned here is the size of the *init()* and *update()* functions

C Interface

C Interface

Object Definition

The structure of PWMDAC object is defined by following structure definition

```
typedef struct {  
    int16 *PwmDacInPointer0; // Input: Pointer to source data output on PWMDAC ch 0  
    int16 *PwmDacInPointer1; // Input: Pointer to source data output on PWMDAC ch 1  
    int16 *PwmDacInPointer2; // Input: Pointer to source data output on PWMDAC ch 2  
    Uint16 PeriodMax;        // Parameter: PWMDAC half period in number of clocks (Q0)  
    void (*init)();           // Pointer to the init function  
    void (*update)();         // Pointer to the update function  
} PWMDAC;
```

```
typedef PWMDAC *PWMDAC_handle;
```

Item	Name	Description	Format	Range(Hex)
Inputs	PwmDacInPointer(x=0,1,2)	These input variables contain the addresses of the desired s/w variables.	N/A	N/A
Outputs	PWMx (x=7,9,11) or EPWMxA (x=4,5,6)	Output signals from the PWM 7-12 pins in EVB on the x2812eZdsp or EPWM4-6 pins on the x2808eZdsp.	N/A	0-3.3 V
PWMDAC parameter	PeriodMax	PWMDAC half period in number of clocks	Q0	8000-7FFF

Special Constants and Data types

PWMDAC

The module definition is created as a data type. This makes it convenient to instance an interface to the PWMDAC driver. To create multiple instances of the module simply declare variables of type PWMDAC.

PWMDAC_handle

User defined Data type of pointer to PWMDAC module

PWMDAC_DEFAULTS

Structure symbolic constant to initialize PWMDAC module. This provides the initial values to the terminal variables as well as method pointers.

Methods

```
void F281X_EV2_PWMDAC_Init(PWMDAC *);  
void F281X_EV2_PWMDAC_Update(PWMDAC *);
```

```
void F280X_PWMDAC_Init(PWMDAC *);  
void F280X_PWMDAC_Update(PWMDAC *);
```

This default definition of the object implements two methods – the initialization and the runtime compute function for PWMDAC generation. This is implemented by means of a function pointer, and the initializer sets this to F281X_EV2_PWMDAC_Init and F281X_EV2_PWMDAC_Update functions for x281x or F280X_PWMDAC_Init and F280X_PWMDAC_Update functions for x280x. The argument to this function is the address of the PWMDAC object.

Module Usage

Instantiation

The following example instances one PWMDAC object
PWMDAC pwmdac1;

Initialization

To Instance pre-initialized objects
PWMDAC pwmdac1 = PWMDAC_DEFAULTS;

Invoking the computation function

pwmdac1.init(&pwmdac1);
pwmdac1.update(&pwmdac1);

Example

The following pseudo code provides the information about the module usage.

```
main()
{
    pwmdac1.PeriodMax = 2500;    // PWM frequency = 30 kHz, clock = 150 MHz
    pwmdac1.PwmDacInPointer0= &pwmdac_ch1_Q15;
    pwmdac1.PwmDacInPointer1= &pwmdac_ch2_Q15;
    pwmdac1.PwmDacInPointer2= &pwmdac_ch3_Q15;

    pwmdac1.init(&pwmdac1);      // Call init function for pwmdac1
}

void interrupt periodic_interrupt_isr()
{
    pwmdac_ch1_Q15 = (int)_IQtoIQ15(variable1_in_IQ);
    pwmdac_ch2_Q15 = (int)_IQtoIQ15(variable2_in_IQ);
    pwmdac_ch3_Q15 = (int)_IQtoIQ15(variable3_in_IQ);

    pwmdac1.update(&pwmdac1);    // Call update function for pwmdac1
}
```

Technical Background

This module converts any s/w variables with Q15 representation into the PWM signals in EVB for x281x by means of calculation the (full) compare values using Timer 3 for generating PWM outputs. For 280x devices, the EPWM4-6 will be used instead. Therefore, the external low-pass filters are necessary to view the actual signal waveforms as seen in Figure 1. The (1st-order) RC low-pass filter can be simply used for filter out the high frequency component embedded in the actual low frequency signals. To select R and C values, its time constant can be expressed in term of cut-off frequency (f_c) as follow:

$$\tau = RC = \frac{1}{2\pi f_c} \quad (1)$$

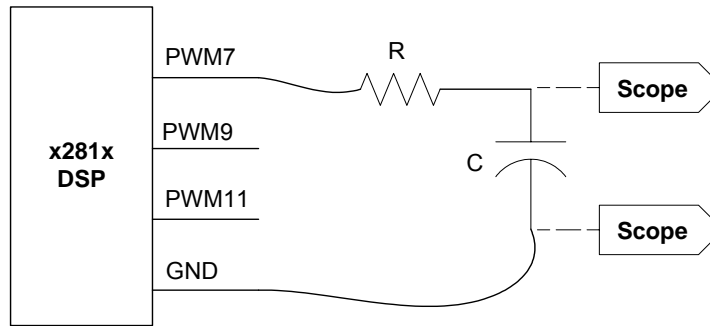


Figure 1: External RC low-pass filter connecting to PWM7 pin in x281x DSP

Note that for 281x PWM8, PWM10, and PWM12 signals are out of phase from PWM7, PWM9, and PWM11 signals, respectively. For 280x, EPWM4B, EPWM5B, and EPWM6B signals are out of phase from EPWM4A, EPWM5A, and EPWM6A signals, respectively.