# Table of contents

1. Overview ........................................................................................................................................ 2
2. Module Properties ......................................................................................................................... 2
3. Module Input and Output Definitions ............................................................................................ 3
   3.1 Module inputs ............................................................................................................................ 3
   3.2 Module outputs .......................................................................................................................... 3
4. Module API Description ................................................................................................................ 4
   4.1 HrPwmDacDrvCnf..................................................................................................................... 4
   4.2 HR_PWM_DAC_INIT .................................................................................................................. 5
   4.3 HRPWM_DAC_DRV .................................................................................................................. 6
5. Usage Example ............................................................................................................................. 7
6. Detailed description ...................................................................................................................... Error! Bookmark not defined.

# Table of Figures

Figure 1. High resolution PWM based DAC .................................................................................... 2
Figure 2. Connecting the high resolution buck converter ................................................................. 7

# Index of Tables

Table 1. HR_PWM_DAC_DRV module dependencies ........................................................................ 2
Table 2. HR_PWM_DAC_DRV module components ........................................................................ 2
Table 3. HR_PWM_DAC_DRV module miscellaneous properties .................................................... 3
Table 4. HR_PWM_DAC_DRV module component files .................................................................. 3
1 Overview

This software module directly controls the EPWM peripherals on the 280x devices to simulate a digital to analog converter, in conjunction with an output filter. It generates appropriate high resolution PWM signals which, when filtered, represent the input values pointed to by the input pointer. This is a useful tool for output of analog values and for visualization.

![Figure 1. High resolution PWM based DAC](image)

2 Module Properties

This section describes module properties, such as compatible devices, components, invocation etc. The HR_PWM_DAC_DRV module has the following dependencies:

<table>
<thead>
<tr>
<th>Module</th>
<th>Dependency</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU dependency</td>
<td>C28x</td>
</tr>
<tr>
<td>Device dependency</td>
<td>x2801 / x2806 / x2808 members only</td>
</tr>
</tbody>
</table>

*Table 1. HR_PWM_DAC_DRV module dependencies*

The HRBUCK_DRV module has the following components:

<table>
<thead>
<tr>
<th>Component</th>
<th>Present</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-based initialization</td>
<td>Yes</td>
</tr>
<tr>
<td>ASM interrupt initialization</td>
<td>Yes</td>
</tr>
<tr>
<td>ASM runtime macro</td>
<td>Yes</td>
</tr>
</tbody>
</table>

*Table 2. HR_PWM_DAC_DRV module components*
The HR_PWM_DAC_DRV module has the following miscellaneous properties:

<table>
<thead>
<tr>
<th>Property name</th>
<th>Property value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiple instance support</td>
<td>Yes</td>
</tr>
<tr>
<td>Reentrant</td>
<td>No</td>
</tr>
<tr>
<td>Accessible from ‘C’ environment</td>
<td>Yes</td>
</tr>
<tr>
<td>Full configuration from ‘C’ environment</td>
<td>Yes</td>
</tr>
<tr>
<td>Input / Output connection</td>
<td>Pointer to signal net.</td>
</tr>
</tbody>
</table>

*Table 3. HR_PWM_DAC_DRV module miscellaneous properties*

### Component files

- `C:\tidcs\DPS_C280x\Vxyz\lib\drvlib280x\src\PWM_HrPwmDacDrvCnf.c`
- `C:\tidcs\DPS_C280x\Vxyz\lib\drvlib280x\include\PWM_DriverMacro.h`

*Table 4. HR_PWM_DAC_DRV module component files*

## 3 Module Input and Output Definitions

### 3.1 Module inputs

<table>
<thead>
<tr>
<th>Input name</th>
<th>Description</th>
<th>Format</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>HRPWM_DAC_InN</td>
<td>Duty cycle control</td>
<td>Pointer to 16-bit fixed point input</td>
<td>Q15: [0, 1] or [0, 32767]</td>
</tr>
<tr>
<td>SFDACN</td>
<td>Scale factor input for HR PWM MEP unit</td>
<td>Pointer to 16-bit fixed point input</td>
<td>Q15: [0, 1] or [0, 32767]</td>
</tr>
</tbody>
</table>

### 3.2 Module outputs

<table>
<thead>
<tr>
<th>Output name</th>
<th>Description</th>
<th>Format</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPWMnA</td>
<td>F280x/C280x PWM output pin</td>
<td>Pulse width modulated output</td>
<td>See device datasheet for electrical specifications.</td>
</tr>
</tbody>
</table>

† The xyz represents the version number directory level. For instance, a 1.00 release would have v100 in its directory path, and v210 would indicate a release 2.10.
4 Module API Description

This module has three executable code components, as described in Table 2. Each of these components is described in this section.

4.1 HrPwmDacDrvCnf

Function Name: HrPwmDacDrvCnf
Prototype: void HrPwmDacDrvCnf(int16 nEPwmModule, int16 Period);
Return value: None.
Preconditions: The following preconditions must be satisfied:
  - The appropriate EPWM module clock must be enabled in the PCLKCR1 register.

The HrPwmDacDrvCnf function is called from the C environment, and performs driver configuration, including the selection of the target EPWM module, and the PWM period. This function should be executed once during the startup process.

- nEPwmModule: Specifies which EPWM module is initialized.
  - **Valid Range**: 1-6, corresponding to EPWM1-6. If EPWM modules 5 or 6 are selected, the high resolution function is not applicable.

- Period: Specifies the PWM period in cycles, corresponding to the high speed peripheral clock.
  - **Valid Range**: 1 to 32767.

Example: Call the HrPwmDacDrvCnf function to use the EPWM1 module as a DAC.

```c
// ePWM1 target, 1000KHz PWM (100 clock period with a 100MHz High speed peripheral clock
HrPwmDacDrvCnf(1, 100);
```
4.2 HR_PWM_DAC_INIT

Function Name: HRPWM_DAC_DRV_INIT

Prototype: HRPWM_DAC_DRV_INIT nEPwmModule

Return value: None.

Preconditions: The following preconditions must be satisfied:
The appropriate EPWM module clock must be enabled in the PCLKCR1 register, and the C language init routine must be called.

This function is the assembler initialization macro, and must be called in addition to the C language initialization routine, for proper operation of the runtime macro routine. This initialization routine must be executed as part of an assembler initialization routine. This macro routine declares variables, initializes variables to known values, and sets up constants for the runtime macro routines.

- nEPwmModule: Specifies which EPWM module is initialized.
  - Valid Range: 1-6, corresponding to EPWM1-6.

Example: Call the HRPWM_DAC_DRV_INIT to initialize EPWM1 module.

```
;-----------------------------------------------
; ISR Initialisation
;-----------------------------------------------
_ISR_Init: HRPWM_DAC_DRV_INIT 1
LRETR
```
4.3 HRPWM_DAC_DRV

Function Name: HRPWM_DAC_DRV
Prototype: HRPWM_DAC_DRV nEPwmModule
Return value: None.
 Preconditions: The following preconditions must be satisfied:

- The appropriate EPWM module clock must be enabled in the PCLKCR1 register.
- C language init routine must be called.
- The ISR initialization macro HRPWM_DAC_DRV_INIT must be instanced in an assembler initialization routine.

This function is the assembler run time macro, and this creates code that forms a bridge between software controllers and the PWM output. This routine writes values into the PWM control registers to control the PWM duty cycle.

- **nEPwmModule**: Specifies which EPWM module is initialized.
  - **Valid Range**: 1-6, corresponding to EPWM1-6.

**Example**: Call the HRPWM_DAC_DRV in an assembler ISR

```assembly
;---------------------------
; Runtime interrupt service routine
;---------------------------
_ISR_Run: CONTEXT_SAVE ;call macro
                       HRPWM_DAC_DRV 1
;---------------------------
EXIT_ISR: ;Interrupt management before exit
;---------------------------
MOV DP,#ETCLR1>>6
MOV @ETCLR1,#0x01 ; Clear EPWM1 Int flag
;---------------------------
; Restore context & return
;---------------------------
CONTEXT_REST
IRET
```
5 Usage Example:

Usage Example:

![Diagram](image)

**Figure 2. Connecting the high resolution buck converter**

Step1. Call the driver configuration function in C (this is one-time pass through code)

```c
HrpwmDacDrvCnf(1, 100);
```

Step2. Instantiate the INIT macro in assembly (this is one-time pass through code)

`; Instantiate the init macro

```assembly
HRPWM_DAC_DRV_INIT 1
```

Step3. Instantiate the run time macro in assembly (this is usually looped or ISR code)

`; “call” the main macro

```assembly
HRPWM_DAC_DRV 1
```

Step4. (optional) Declare “Signal Nets” to “connect” the module to in “C”

```c
int16 Net1;
```

`; Note Net1 can be simply a global integer variable

Step5. Declare the module “Terminal pointers” in “C”

```c
// HRBUCK_DRV terminal pointers, external references
extern int16 *PWMDAC_InAn, *PWMDAC_InAn, ;
```

Step6. “Connect” the module terminals to the Signal Nets in “C”.

```c
// HRBUCK_DRV connections
    PWMDAC_InAn = &Net1;
```

`; Note this can be done once during init, or dynamically during
`; run time operation, i.e. module connections can be
`; re-configured to other Nets as required by the application.