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 HrFullBdgUnipolar2DrvCnf ..............................................................................................................4
  4.2 HRFULLBDG_UNIPOLAR2_DRV_INIT ..........................................................................................5
  4.3 HRFULLBDG_UNIPOLAR2_DRV ...................................................................................................6
5 Usage Example: ...................................................................................................................................7
6 Detailed Description .............................................................................................................................8

Table of Figures

Figure 1. High resolution full-bridge converter, unipolar2 PWM driver module ............................................2
Figure 2. Connecting the high resolution full-bridge converter using unipolar2 PWM scheme ....................7
Figure 3. High resolution full-bridge converter ..............................................................................................8
Figure 4. PWM generation with the F280x EPWM modules (polarity = 0) ...................................................8
Figure 5. PWM generation with the F280x EPWM modules (polarity = 1) ...................................................9

Index of Tables

Table 1. HRFULLBDG_UNIPOLAR2_DRV module dependencies ..............................................................2
Table 2. HRFULLBDG_UNIPOLAR2_DRV module components ...............................................................2
Table 3. HRFULLBDG_UNIPOLAR2_DRV module miscellaneous properties .........................................3
Table 4. HRFULLBDG_UNIPOLAR2_DRV module component files ........................................................3
1 Overview

This software module directly controls the EPWM peripherals on the 280x devices. It generates appropriate High resolution PWM signals by means of unipolar scheme to control a full-bridge converter using two EPWM1/2 modules. This module forms the interface between the control software and the device PWM pins. It provides duty cycle control from 0%-100% and polarity variable, determining either negative or positive output voltage. It is also geared toward application using high frequency PWM (20kHz – 2MHz).

![Diagram: High resolution full-bridge converter, unipolar2 PWM driver module]

2 Module Properties

This section describes module properties, such as compatible devices, components, invocation etc. The HRFULLBDG_UNIPOLAR2_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. HRFULLBDG_UNIPOLAR2_DRV module dependencies*

The HRFULLBDG_UNIPOLAR2_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. HRFULLBDG_UNIPOLAR2_DRV module components*
The HRFULLBDG_UNIPOLAR2_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 (limited by number of physical EPWM modules on a given device).</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. HRFULLBDG_UNIPOLAR2_DRV module miscellaneous properties*

Component files

C:\tidcs\DPS_C280x\V\xyz\lib\drvlib280x\src\PWM_HrFullBdgUnipolar2DrvCnf.c
C:\tidcs\DPS_C280x\V\xyz\lib\drvlib280x\include\PWM_DriverMacro.h

*Table 4. HRFULLBDG_UNIPOLAR2_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>In</td>
<td>Duty cycle control (0%, 100%)</td>
<td>Pointer to 16-bit fixed point input data</td>
<td>Q15: [0, 1] or [0, 32767]</td>
</tr>
<tr>
<td>Polarity</td>
<td>Polarity control 0 = positive, 1 = negative</td>
<td>Pointer to 16-bit fixed point input data</td>
<td>Q0 [0 or 1]</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>EPWM1A/1B/2A/2B</td>
<td>F280x/C280x PWM output pins</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 HrFullBdgUnipolar2DrvCnf

Function Name: HrFullBdgUnipolar2DrvCnf
Prototype: void HrFullBdgUnipolar2DrvCnf(int16 period);
Return value: None.
Preconditions: The following preconditions must be satisfied:
The appropriate EPWM modules clock must be enabled in the PCLKCR1 register.

The HrFullBdgUnipolar2DrvCnf function is called from the C environment, and performs driver configuration of the target EPWM modules and the PWM period. This function should be executed once during the startup process.

- **period**: Specifies the PWM period in cycles, corresponding to the period of EPWM time-base clock ($T_{TBCLK}$). Notice that the PWM is in symmetrical mode (i.e., up-down count).
  - **Valid Range**: 1 to 32767 TBCLK cycles.
    
    \[
    PWM \text{ Frequency(}Hz) = \frac{1}{T_{TBCLK} \times 2 \text{ Period}}
    \]

Example: Call the HrFullBdgUnipolar2DrvCnf function to initialize EPWM modules.

    //----------------------------------------------------------------------
    // for 30KHz PWM and 1 usec dead-time (based on TBCLK frequency of
    // 100 MHz)
    //----------------------------------------------------------------------
    HrFullBdgUnipolar2DrvCnf(1667);
4.2 HRFULLBDG_UNIPOLAR2_DRV_INIT

Function Name: HRFULLBDG_UNIPOLAR2_DRV_INIT

Prototype: HRFULLBDG_UNIPOLAR2_DRV_INIT

Return value: None.

Preconditions: The following preconditions must be satisfied:

The appropriate EPWM modules 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.

Example: Call the HRFULLBDG_UNIPOLAR2_DRV_INIT to initialize EPWM modules.

;;;;;;ISR Initialisation
;;;;;;ISR_Init: HRFULLBDG_UNIPOLAR2_DRV_INIT
LRETR
4.3 HRFULLBDG_UNIPOLAR2_DRV

**Function Name:** HRFULLBDG_UNIPOLAR2_DRV  
**Prototype:** HRFULLBDG_UNIPOLAR2_DRV  
**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 HRFULLBDG_UNIPOLAR2_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 outputs. This routine writes values into the PWM control registers to control the PWM duty cycle.

**Example:** Call the HRFULLBDG_UNIPOLAR2_DRV in an assembler ISR

```
;---------------------------------------------------------
; Runtime interrupt service routine
;---------------------------------------------------------
_ISR_Run: CONTEXT_SAVE ;call macro
                     HRFULLBDG_UNIPOLAR2_DRV
;---------------------------------------------------------
EXIT_ISR: ;Interrupt management before exit
;---------------------------------------------------------
            MOVW  DP,#ETCLR1>>6
            MOV  @ETCLR1,#0x01 ; Clear EPWM1 Int flag
;---------------------------------------------------------
; Restore context & return
;---------------------------------------------------------
CONTEXT_REST
IRET
```
5 Usage Example:

Usage Example:

Figure 2. Connecting the high resolution full-bridge converter using unipolar2 PWM scheme

Step 1. Call the driver configuration function in C (this is one-time pass through code)

    HRFullBdgUnipolar2DrvCnf(1667);

Step 2. Instantiate the INIT macro in assembly (this is one-time pass through code)

    ; Instantiate the init macro
    HRFULLBDG_UNIPOLAR2_DRV_INIT

Step 3. Instantiate the run time macro in assembly (this is usually looped or ISR code)

    ; "call" the main macro
    HRFULLBDG_UNIPOLAR2_DRV

Step 4. (optional) Declare “Signal Nets” to “connect” the module to in “C”

    // Note: Net1 can be simply a global integer variable
    int16  Net1;
    Uint16  Polarity1;

Step 5. Declare the module “Terminal pointers” in “C”

    // HRFULLBDG_UNIPOLAR2_DRV terminal pointers, external references
    extern int16   *HRFullBdg_UNI2_In;
    extern Uint16   *HRFullBdg_UNI2_Polarity;

Step 6. “Connect” the module terminals to the Signal Nets in “C”.

    // HRFULLBDG_UNIPOLAR2_DRV connections
    HRFullBdg_UNI2_In = &Net1;
    HRFullBdg_UNI2_Polarity = &Polarity1;

    // 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.
6 Detailed Description

Figure 3. High resolution full-bridge converter

Figure 4. PWM generation with the F280x EPWM modules (polarity = 0)
Figure 5. PWM generation with the F280x EPWM modules (polarity = 1)
**HRFULLBDG_UNIPOLAR2_DRV**

(Symmetrical – Up-Down count)

<table>
<thead>
<tr>
<th>HSPCLK =</th>
<th>100</th>
<th>1.00E+03 MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPH3IL Freq =</td>
<td>30</td>
<td>3.00E+01 KHz</td>
</tr>
</tbody>
</table>

**Duty cycle calculation table**

<table>
<thead>
<tr>
<th>per unit</th>
<th>Duty(%)</th>
<th>MPH3IL_duty</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Period count (dec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Q15 (Dec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Compare Count (dec)</td>
</tr>
<tr>
<td>1.00</td>
<td>100%</td>
<td>32767</td>
</tr>
<tr>
<td>0.90</td>
<td>90%</td>
<td>29490</td>
</tr>
<tr>
<td>0.80</td>
<td>80%</td>
<td>26214</td>
</tr>
<tr>
<td>0.70</td>
<td>70%</td>
<td>22937</td>
</tr>
<tr>
<td>0.60</td>
<td>60%</td>
<td>19660</td>
</tr>
<tr>
<td>0.50</td>
<td>50%</td>
<td>16384</td>
</tr>
<tr>
<td>0.40</td>
<td>40%</td>
<td>13107</td>
</tr>
<tr>
<td>0.30</td>
<td>30%</td>
<td>9830</td>
</tr>
<tr>
<td>0.20</td>
<td>20%</td>
<td>6553</td>
</tr>
<tr>
<td>0.10</td>
<td>10%</td>
<td>3277</td>
</tr>
<tr>
<td>0.00</td>
<td>0%</td>
<td>0</td>
</tr>
<tr>
<td>-0.10</td>
<td>0%</td>
<td>62259</td>
</tr>
<tr>
<td>-0.20</td>
<td>0%</td>
<td>58982</td>
</tr>
<tr>
<td>-0.30</td>
<td>0%</td>
<td>55706</td>
</tr>
<tr>
<td>-0.40</td>
<td>0%</td>
<td>52429</td>
</tr>
<tr>
<td>-0.50</td>
<td>0%</td>
<td>49152</td>
</tr>
<tr>
<td>-0.60</td>
<td>0%</td>
<td>45875</td>
</tr>
<tr>
<td>-0.70</td>
<td>-70%</td>
<td>42598</td>
</tr>
<tr>
<td>-0.80</td>
<td>-80%</td>
<td>39322</td>
</tr>
<tr>
<td>-0.90</td>
<td>0%</td>
<td>36045</td>
</tr>
<tr>
<td>-1.00</td>
<td>0%</td>
<td>32768</td>
</tr>
</tbody>
</table>

Note: Negative duty cycle not defined and will give 0% duty.