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 HrBuckDrvCnf ........................................................................................................ 4
   4.2 HrBuckDrvCnfV2 .................................................................................................. 5
   4.3 HRBUCK_DRV_INIT ............................................................................................ 6
   4.4 HRBUCK_DRV ..................................................................................................... 7
5 Usage Example ......................................................................................................... 8
6 Detailed description .................................................................................................. 8

Table of Figures
Figure 1. High resolution buck converter, PWM driver module .................................................. 2
Figure 2. Connecting the high resolution buck converter ............................................................. 8
Figure 3. High resolution buck converter ................................................................................ 9
Figure 4. PWM generation with the F280x EPWM module ....................................................... 9

Index of Tables
Table 1. HRBUCK_DRV module dependencies ......................................................................... 2
Table 2. HRBUCK_DRV module components .......................................................................... 2
Table 3. HRBUCK_DRV module miscellaneous properties ....................................................... 3
Table 4. HRBUCK_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 to control a Buck converter using only a single EPWMx module. This module forms the interface between the control software and the device PWM pins. It provides duty cycle control from 0-100%, and is geared toward application using high frequency PWM (200kHz – 2MHz).

![Diagram](image)

*Figure 1. High resolution buck converter, PWM driver module*

2 Module Properties

This section describes module properties, such as compatible devices, components, invocation etc. The HRBUCK_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. HRBUCK_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. HRBUCK_DRV module components*

The HRBUCK_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>
</tbody>
</table>
Table 3. HRBUCK_DRV module miscellaneous properties

<table>
<thead>
<tr>
<th>Component files</th>
</tr>
</thead>
<tbody>
<tr>
<td>C:\tidcs\DPS_C280x\Vxyz\lib\drvlib280x\src\PWM_HRBuckDrvCnf.c</td>
</tr>
<tr>
<td>C:\tidcs\DPS_C280x\Vxyz\lib\drvlib280x\include\PWM_DriverMacro.h</td>
</tr>
</tbody>
</table>

Table 4. HRBUCK_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>Duty</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>
</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 HrBuckDrvCnf

Function Name: HrBuckDrvCnf

Prototype: void HrBuckDrvCnf(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.

- Valid Range: 1-6, corresponding to EPWM1-6.

Note: Only EPWM 1-4 allow for high resolution operation on the TMS320F2801, TMS320F2806, and TMS320F2808. EPWM5-6 will operate with standard resolution (1/clock cycle).

- Valid Range: 1 to 32767 TBCLK cycles.

\[
PWM\text{ Frequency} = \frac{HSPCLK(High\text{ Speed Per CLK})}{Period}
\]

Example: Call the HrBuckDrvCnf function to initialize EPWM1 module.

```c
// --------------------
// 1000KHz PWM (100 clock period with a 100MHz High speed peripheral clock
// --------------------
HrBuckDrvCnf(1, 100);
```
HrBuckDrvCnfV2

Function Name:        HrBuckDrvCnfV2
Prototype:            void HrBuckDrvCnfV2 (int16 nEPwmModule,
                          int16 period,
                          int16 mode,
                          int16 phase);

Return value:        None.
Preconditions:       The following preconditions must be satisfied:

                      The appropriate EPWM module clock must be enabled in the PCLKCR1
                      register.

The HrBuckDrvCnfV2 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. This function allows the HRBUCK
driver to be configured with additional flexibility.

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

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

- **Mode**: Specifies whether the EPWM module is configured as a master or as a slave.
  - **1**: EPWM module is configured as a master.
  - **0**: EPWM module is configured as a slave.

- **Phase**: Phase offset from the master module, applicable only if the module is in slave mode.

**Example**: Call the HrBuckDrvCnfV2 function to initialize EPWM1 module as a master, and
EPWM2 as a slave with zero phase.

```c
//-------------------------------------
// ePWM1 target, 1000KHz PWM, master mode
// (100 clock period with a 100MHz High speed peripheral clock
//-------------------------------------
HrBuckDrvCnfV2(1, 100, 1, 0); // ePWM1 Master
HrBuckDrvCnfV2(2, 100, 0, 0); // ePWM2 Slave
```
4.2 HRBUCK_DRV_INIT

Function Name: HRBUCK_DRV_INIT

Prototype: HRBUCK_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 HRBUCK_DRV_INIT to initialize EPWM1 module.

```c
ISR Initialisation

_ISR_Init: HRBUCK_DRV_INIT 1
LRETR
```
4.3 HRBUCK_DRV

Function Name: HRBUCK_DRV

Prototype: HRBUCK_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 HRBUCK_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 HRBUCK_DRV in an assembler ISR

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

5   Usage Example:

Figure 2. Connecting the high resolution buck converter

Step1. Call the driver configuration function in C (this is one-time pass through code)
   HrBuckDrvCnf(1, 200);
   HrBuckDrvCnf(2, 200);

Step2. Instantiate the INIT macro in assembly (this is one-time pass through code)
   ; Instantiate the init macro
   HRBUCK_DRV_INIT 1
   HRBUCK_DRV_INIT 2

Step3. Instantiate the run time macro in assembly (this is usually looped or ISR code)
   ; “call” the main macro
   HRBUCK_DRV 1
   HRBUCK_DRV 2

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

   // Note: Net1, Net2 can be simply a global integer variable
   int16   Net1, Net2;

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

   // HRBUCK_DRV terminal pointers, external references
   extern int16   *HRBUCK_Duty1, *HRBUCK_Duty2;

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

   // HRBUCK_DRV connections
   HRBUCK_Duty1 = &Net1;
   HRBUCK_Duty2 = &Net2;

   // 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 buck converter

Figure 4. PWM generation with the F280x EPWM module.
HRBUCK_DRV

(Asymmetrical - Up count)

HSPCLK = 100 MHz
MPH3IL Freq = 100 KHz

TBCLK = Sysclk * Pre-scale

**Duty cycle calculation table**

<table>
<thead>
<tr>
<th>Duty(%)</th>
<th>Q15 (Dec)</th>
<th>Q15 (Hex)</th>
<th>MPH3IL_duty</th>
<th>Period count (dec)</th>
<th>Period count (hex)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>32767</td>
<td>7FFF</td>
<td>Compare Count (dec)</td>
<td>500</td>
<td>Compare Count (hex)</td>
</tr>
<tr>
<td>0.90</td>
<td>29490</td>
<td>7332</td>
<td></td>
<td>450</td>
<td>01C2</td>
</tr>
<tr>
<td>0.80</td>
<td>26214</td>
<td>6665</td>
<td></td>
<td>400</td>
<td>0190</td>
</tr>
<tr>
<td>0.70</td>
<td>22937</td>
<td>5998</td>
<td></td>
<td>350</td>
<td>015E</td>
</tr>
<tr>
<td>0.60</td>
<td>19660</td>
<td>4CCC</td>
<td></td>
<td>300</td>
<td>012C</td>
</tr>
<tr>
<td>0.50</td>
<td>16384</td>
<td>3FFF</td>
<td></td>
<td>250</td>
<td>00FA</td>
</tr>
<tr>
<td>0.40</td>
<td>13107</td>
<td>3332</td>
<td></td>
<td>200</td>
<td>00C8</td>
</tr>
<tr>
<td>0.30</td>
<td>9830</td>
<td>2666</td>
<td></td>
<td>150</td>
<td>0096</td>
</tr>
<tr>
<td>0.20</td>
<td>6553</td>
<td>1999</td>
<td></td>
<td>100</td>
<td>0064</td>
</tr>
<tr>
<td>0.10</td>
<td>3277</td>
<td>0CCC</td>
<td></td>
<td>50</td>
<td>0032</td>
</tr>
<tr>
<td>0.00</td>
<td>0</td>
<td>0000</td>
<td></td>
<td>50</td>
<td>0000</td>
</tr>
<tr>
<td>-0.10</td>
<td>62259</td>
<td>F333</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.20</td>
<td>58982</td>
<td>E666</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.30</td>
<td>55706</td>
<td>D999</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.40</td>
<td>52429</td>
<td>CCCC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.50</td>
<td>49152</td>
<td>C000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.60</td>
<td>45875</td>
<td>B333</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.70</td>
<td>42598</td>
<td>A666</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.90</td>
<td>36045</td>
<td>8CCC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1.00</td>
<td>32768</td>
<td>8000</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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