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 ........................................................................................................... Error! Bookmark not defined.
  4.2 HrBuckDrvCnfV2 ..................................................................................................... Error! Bookmark not defined.
  4.3 HRBUCK_DRV_INIT .............................................................................................. 5
  4.4 HRBUCK_DRV ....................................................................................................... Error! Bookmark not defined.
5 Usage Example ............................................................................................................ 7
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 ..................................................... Error! Bookmark not defined.
Figure 3. High resolution buck converter ......................................................................... 8
Figure 4. PWM generation with the F280x EPWM module .................................................. 8

Index of Tables

Table 1. HRBUCK_DRV module dependencies ................................................................. 2
Table 2. HRBUCK_DRV module components .................................................................. 2
Table 3. HRBUCK_DRV module miscellaneous properties ................................................ 2
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 PWM signals to control a 3 phase interleaved buck by using 3 EPWM modules synchronized together with phase offset. This module forms the interface between the control software and the device PWM pins.

![Figure 1. Multiphase buck converter, PWM driver module](image)

2 Module Properties

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

The MPIL_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>No</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. MPIL_DRV module miscellaneous properties*
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</td>
<td>Pointer to 16-bit fixed</td>
<td>Q15: [0, 1] or [0, 32767]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>point input data</td>
<td></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>
<tr>
<td>EPWMnB</td>
<td>F280x/C280x PWM output pin</td>
<td>Pulse width modulated output.</td>
<td>See device datasheet for electrical specifications.</td>
</tr>
<tr>
<td>. . .</td>
<td>. . .</td>
<td>. . .</td>
<td>. . .</td>
</tr>
<tr>
<td>EPWM(n+N)A</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 MPILDrvCnf

Function Name: MPILDrvCnf
Prototype: void MPILDrvCnf(int16 nEPwmModule, int16 nChCount, 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 HrBuckDrvCnf 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.

- nChCount: Specifies how many EPWM modules are initialized.
  - **Valid Range:** (7-nEPwmModule).

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

**Example:** Call the MPILDrvCnf function to initialize EPWM1-3 modules.

```c
//ePWM1-3 target, 1000KHz PWM (100 clock period with a 100MHz High speed peripheral clock
MPILDrvCnf(1, 3, 100);```

4.2 HRBUCK_DRV_INIT

Function Name: MPIL_DRV_INIT

Prototype: MPIL_DRV_INIT nEPwmModule, nChCount

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.

- nChCount: Specifies how many EPWM modules are initialized.
  - Valid Range: (7-nEPwmModule).

Example: Call the MPIL_DRV_INIT to initialize EPWM1-3 module.

```c
;---------------------------------------------------------
; ISR Initialisation
;---------------------------------------------------------
_ISR_Init: HRBUCK_DRV_INIT 1,3
LRETR
```
4.3 MPIL_DRV

Function Name: MPIL_DRV
Prototype: MPIL_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 MPIL_DRV 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.
- nChCount: Specifies how many EPWM modules are initialized.
  - **Valid Range:** (7-nEPwmModule).

**Example:** Call the MPIL_DRV in an assembler ISR to implement runtime code for EPWM modules 1-3.

```asm
;---------------------------------------------------------
; Runtime interrupt service routine
;---------------------------------------------------------
_ISR_Run: CONTEXT_SAVE   ;call macro
      MPIL_DRV 1,3
;---------------------------------------------------------
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:

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

PWM_MPILDrvCnf();

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

; Instantiate the init macro

MPIL_DRV_INIT nEPWMModule, nChCount

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

; “call” the main macro

MPIL_DRV nEPWMModule, nChCount

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

int Net1;

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

// MPH3IL_DRV terminal pointers, external references
extern int *MPIL_Duty;

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

// MPIL_DRV connections

MPIL_Duty = &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.
6 Detailed description

![Figure 2. High resolution buck converter](image)

![Figure 3. PWM generation with the F280x EPWM module.](image)
**MPIL_DRV**

(symmetrical UpDown count)

<table>
<thead>
<tr>
<th>HSPCLK = 100</th>
<th>1.00E+06 MHz</th>
<th>TBCLK = Sysclk * Pre-scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPH3IL Freq = 100</td>
<td>1.00E+03 KHz</td>
<td></td>
</tr>
</tbody>
</table>

### Duty cycle calculation table

<table>
<thead>
<tr>
<th>Duty cycle calculation table</th>
<th>MPIL_duty</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Per unit</strong></td>
<td><strong>Duty(%)</strong></td>
</tr>
<tr>
<td>1.00</td>
<td>100%</td>
</tr>
<tr>
<td>0.90</td>
<td>90%</td>
</tr>
<tr>
<td>0.80</td>
<td>80%</td>
</tr>
<tr>
<td>0.70</td>
<td>70%</td>
</tr>
<tr>
<td>0.60</td>
<td>60%</td>
</tr>
<tr>
<td>0.50</td>
<td>50%</td>
</tr>
<tr>
<td>0.40</td>
<td>40%</td>
</tr>
<tr>
<td>0.30</td>
<td>30%</td>
</tr>
<tr>
<td>0.20</td>
<td>20%</td>
</tr>
<tr>
<td>0.10</td>
<td>10%</td>
</tr>
<tr>
<td>0.00</td>
<td>0%</td>
</tr>
<tr>
<td>-0.10</td>
<td>0%</td>
</tr>
<tr>
<td>-0.20</td>
<td>0%</td>
</tr>
<tr>
<td>-0.30</td>
<td>0%</td>
</tr>
<tr>
<td>-0.40</td>
<td>0%</td>
</tr>
<tr>
<td>-0.50</td>
<td>0%</td>
</tr>
<tr>
<td>-0.60</td>
<td>0%</td>
</tr>
<tr>
<td>-0.70</td>
<td>0%</td>
</tr>
<tr>
<td>-0.90</td>
<td>0%</td>
</tr>
<tr>
<td>-1.00</td>
<td>0%</td>
</tr>
</tbody>
</table>

*Note: Negative duty cycle not defined.*