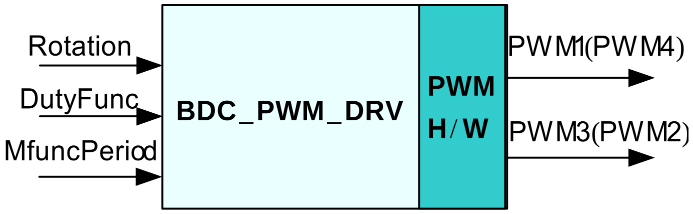
BDC_PWM_DRV — *Brushed DC-Motor PWM Driver*

Description

This module generates the PWM signals for brushes DC motor drive. The topology of converter is H-bridge where two upper switches are controlled by PWM signals. The rotational direction of motor can be changed by the input *Rotation*. The duty cycle values for the PWM outputs are also determined by the input *DutyFunc*.



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: f281xbdcpwm.c, f281xbdcpwm.h (for x281x)
f280xbdcpwm.c, f280xbdcpwm.h (for x280x)

IQmath library files for C: N/A

Item	**C version**	**Comments**
Code Size□ (x281x/x280x)	79/252 words	
Data RAM	0 words•	
xDAIS ready	No	
XDAIS component	No	IALG layer not implemented
Multiple instances	Yes	
Reentrancy	Yes	

• Each pre-initialized PWMGEN structure consumes 9 words in the data memory

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

C Interface

Object Definition

The structure of PWMGEN object is defined by following structure definition

```
typedef struct {   Uint16 Rotation;        // Input: 0 = PWM1&4 enabled, 1 = PWM2&3 enabled (Q0)
                   int16 MfuncPeriod;      // Input: Period scaler (Q15)
                   int16 DutyFunc;         // Input: PWM period modulation input (Q15)
                   Uint16 PeriodMax;       // Parameter: Maximum period (Q0)
                   Uint16 PwmActive;       // Parameter: 0 = PWM active low, 1 = PWM active high
                   void (*init)();         // Pointer to the init function
                   void (*update)();       // Pointer to the update function
               } PWMGEN ;

typedef PWMGEN *PWMGEN_handle;
```

Item	Name	Description	Format	Range(Hex)
Inputs	Rotation	Rotational direction of motor	Q0	0 or 1
	MfuncPeriod	Duty ratio of the PWM outputs	Q15	8000-7FFF
	DutyFunc	PWM period modulation input	Q15	8000-7FFF
Outputs	PWMx (x=1,2,3,4)	Output signals from the 4 PWM pins in EVA on the x2812eZdsp.	N/A	0-3.3 V
PWMGEN parameter	PeriodMax	PWM Period in CPU clock cycles	Q0	8000-7FFF
	PwmActive	0 = PWM active low 1 = PWM active high	Q0	0 or 1

Special Constants and Data types

PWMGEN

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

PWMGEN_handle

User defined Data type of pointer to PWMGEN module

PWMGEN_DEFAULTS

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

Methods

void F281X_EV1_BDC_PWM_Init(PWMGEN *);
void F281X_EV1_BDC_PWM_Update(PWMGEN *);

void F280X_BDC_PWM_Init(PWMGEN *);
void F280X_BDC_PWM_Update(PWMGEN *);

This default definition of the object implements two methods – the initialization and the runtime compute function for PWMGEN generation. This is implemented by means of a function pointer, and the initializer sets this to F281X_EV1_BDC_PWM_Init and F281X_EV1_BDC_PWM_Update functions for x281x or F280X_BDC_PWM_Init and F280X_BDC_PWM_Update functions for x280x. The argument to this function is the address of the PWMGEN object.

Module Usage

Instantiation

The following example instances one PWMGEN object

```
PWMGEN  pwm1;
```

Initialization

To Instance pre-initialized objects

```
PWMGEN pwm1 = PWMGEN_DEFAULTS;
```

Invoking the computation function

```
pwm1.init(&pwm1);
pwm1.update(&pwm1);
```

Example

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

```
main()
{

        pwm1.PeriodMax = 7500;          // PWM frequency = 20 kHz, clock = 150 MHz
        pwm1.init(&pwm1);               // Call init function for pwm1

}

void interrupt periodic_interrupt_isr()
{

        pwm1.Rotation = 1;                              // rotation is either 0 or 1
        pwm1.DutyFunc = (int)_IQtoIQ15(d_func1);         // d_func1 is in GLOBAL_Q
        pwm1.update(&pwm1);                             // Call update function for pwm1


}
```

Technical Background

Figure 1 shows the simplified topology of H-bridge dc/dc converter used to drive a brushed DC motor. This kind of converter topology allows four-quadrant operation. There are several switching strategies that might be used with a four-quadrant operation. For this PWM driver, the motor is driven by a pair of switches depending on the rotation input. For example, when the rotation input is set to 0, switches Q1 and Q4 are active whereas switches Q2 and Q3 are turned off. When the rotation input is set to 1, switches Q2 and Q3 are active whereas switches Q1 and Q4 are turned off. The averaging dc voltage across the brushed DC motor is controlled by means of changing the duty cycle of switches Q1 (for Rotation = 0) or Q3 (for Rotation = 1). Table 1 summarizes the switching scheme for the brushed DC motor drive.

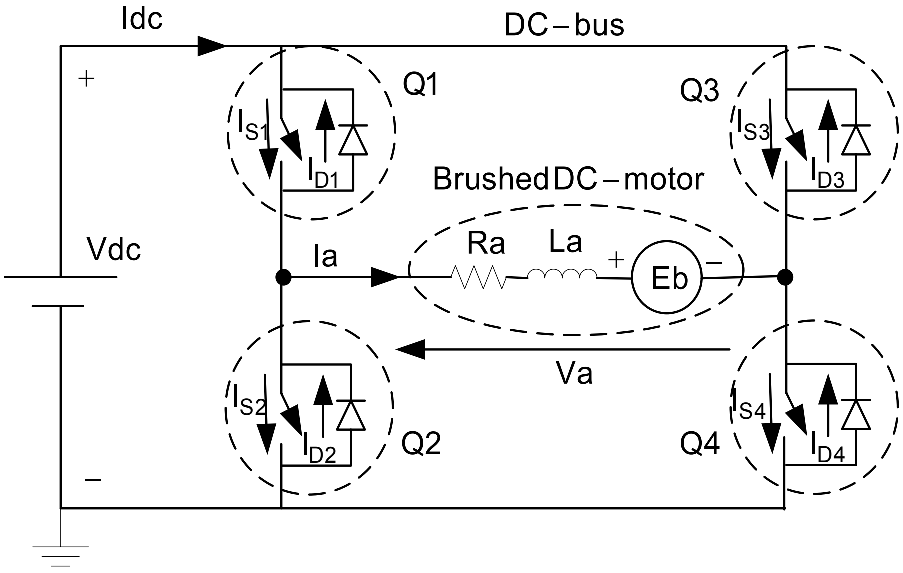


Figure 1: H-bridge converter for a brushed DC motor drive

	Q1	Q2	Q3	Q4
Rotation = 0	**PWM**	OFF	OFF	**ON**
Rotation = 1	OFF	**ON**	**PWM**	OFF

Table 1: Switching scheme for a brushed DC motor drive using H-bridge converter