TN0700x

PWM duel edge driver for LPC23xx, LPC24xx

Rev 0.0— 22-06-2007 Technical note

1. Introduction

The LPC23xx and LPC24xx have PWM with dual edge mode. This can be used to generate 3 or 6 (only LPC24xx) dual edge aligned PWM outputs for motor control or other applications. This Tech. note describes a driver that can be used for this kind of applications.

2. PWM

2.1 Modes

The LPC23xx have one PWM and the LPC24xx have two PWM blocks with 6 PWM outputs in single edge mode and 3 PWM outputs in dual edge mode. The two PWM blocks of the LPC24xx can be synchronized with each other. In single edge mode the PWM outputs will be set to “1” at the timer overflow and will be reset at a match of the timer with the corresponding match register. In dual edge mode only PWM output 2, 4, 6 are active as output. The match registers of the other not used outputs are used for setting the (active) outputs and the match registers of the active outputs are used for clearing the (active) outputs. So match register 1 and 2 are used for PWM output 2, match registers 3,4 for PWM output 4 and match registers 5,6 for PWM output 6 . The combination of two match registers and one output is called (PWM) channel in the document.

The mode has to be set in PWMxPCR register(s). For synchronization of both PWM blocks bit 4 in register PWM0TCR has to be cleared (only for the LPC24xx).

2.2 Pinning

The PWM outputs of the LPC23xx and LPC24xx have to be connected to a port pin. For one PWM output two possibilities are implemented. This can be selected with PINSELX registers. You find in the user manual more detailed information how to establish such a connection.

2.3 Invert output

In some cases a PWM output has to be inverted. This is accomplished in the driver by changing the set and clear values of the corresponding match registers of the channel. One of the parameters in the function PWM_doubleEdge_on and PWM_doubleEdge_off is “invert”, “not invert”.

2.4 Switching off all outputs

Channel number 0 of function PWM_doubleEdge_off is used for switching off all outputs. For non inverting output this will be “0” For inverting output this will be around Vdd. The other channel numbers are for switching of the specific channel.

NXP founded by Philips

3. Software architecture

3.1 Level 0

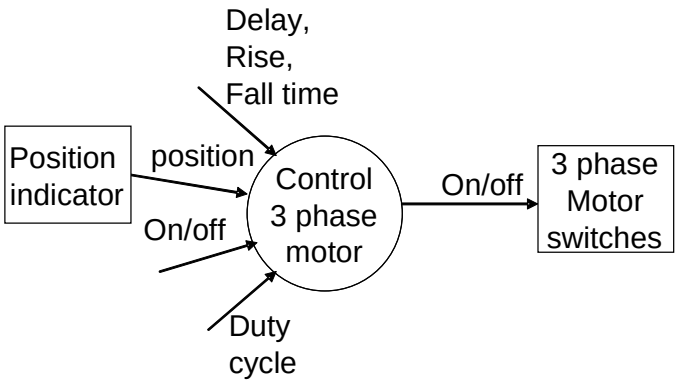
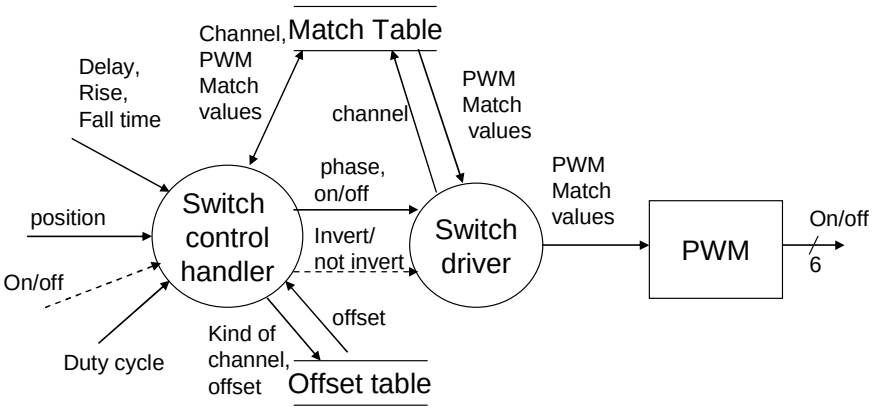


Fig 1

3.2 Level 1(control 3 phase motor)

Control 3 phase motor



The phase switch sequence is determined by in switch driver

Fig 2

3.3 Level 2 (Switch control handler)

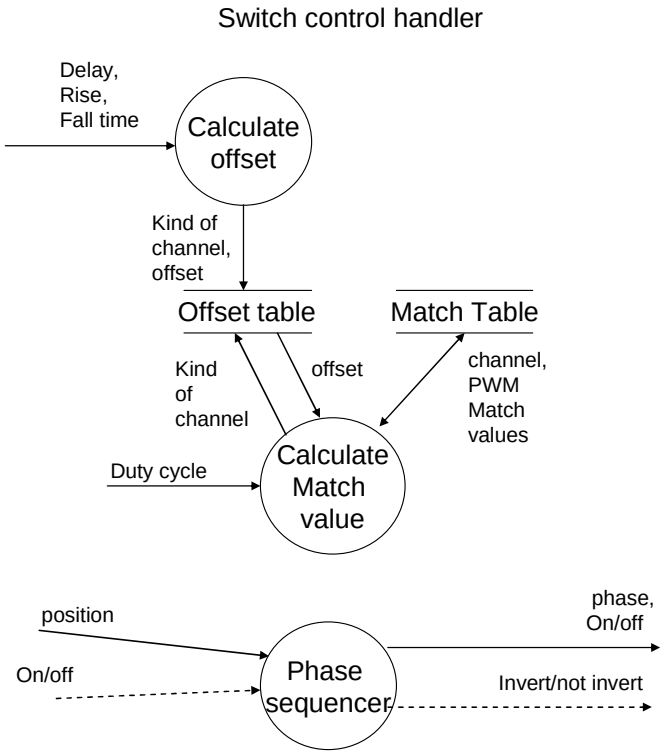


Fig 3

3.4 State diagram

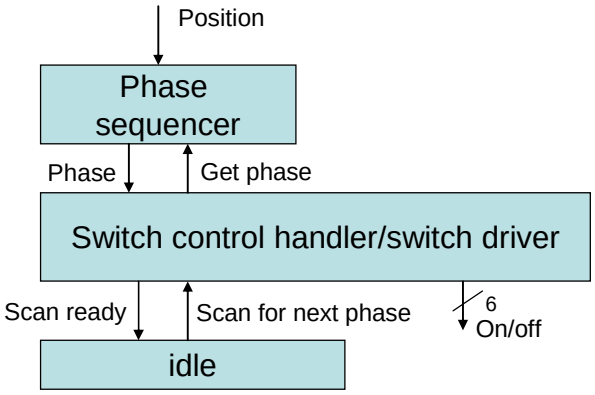


Fig 4

4. Software implementation

4.1 Calculation offset

For changing the duty cycle an offset table is used to store the offset values that can be used for calculation of the values for the match registers. The stored values are: Upper switch set time, Upper switch clear time, Lower switch set time and Lower switch clear time offset. The values are: delay + rise time, delay + rise time, delay and delay + fall time. To Upper switch clear time and Lower clear time the duty cycle time has to be added to get the wave form for switch an upper and lower switch (FET).

Function PWM_set_deadtime is calculating and storing the offset.

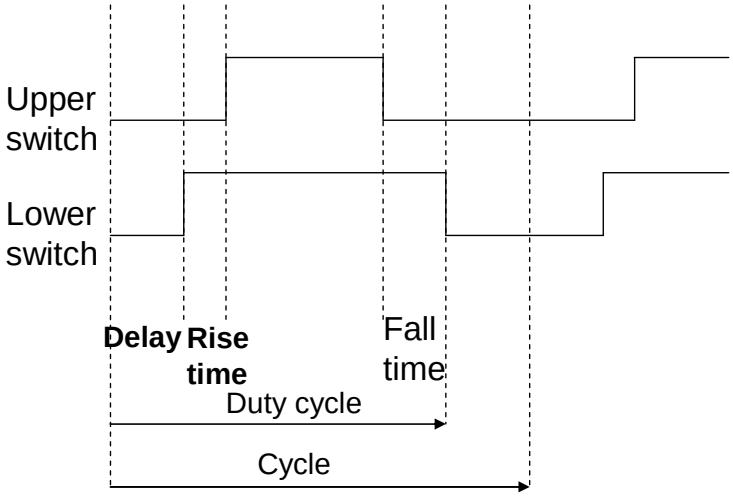


Fig 6

4.2 Calculation match values

The functions Set_duty_cycle and Set_cycle calculate the match values for the channels and put it in a Match_table. Table(0) contains the Match 0 register value, used for the cycle time. The other indexes are used for set and clear values per channel. Function Set_duty_cycle checks the duty cycle value and calculates the set and clear values for upper and lower switches and puts the values in the Match table. The functions PWM_doubleEdge_on and PWM_doubleEdge_off are using the match table for switching the channels.

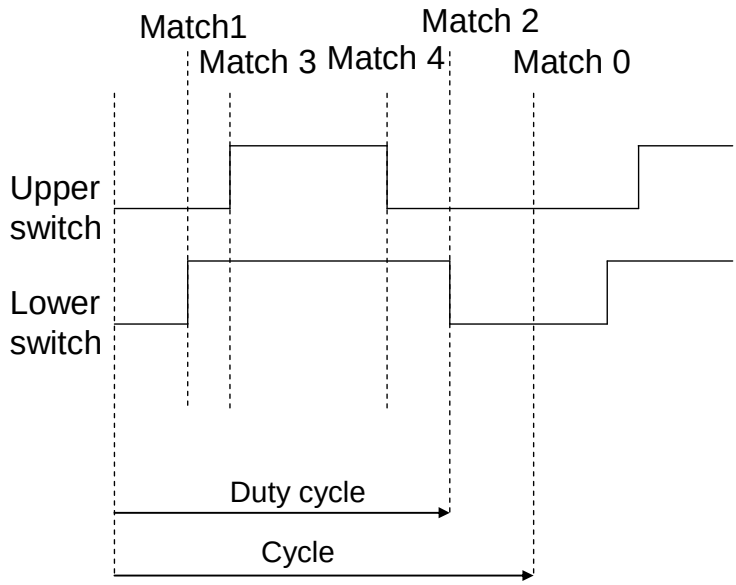


Fig 7

4.3 Switch driver

The function PWM_doubleEdge_on is putting the normal or inverted set, clear values specified in the match table in the Match registers of the PWM block for the specified channel.

The function PWM_doubleEdge_off is putting the normal or inverted “off” values in the Match registers of the PWM block for the specific channel.

The values become active (in PWM block) after the function PWM_activate_match() has been called. The advantage of this approach is that changes of channel switch will be active the same time after changing the match registers. Activating the same time cannot be accomplished completely because there is an instruction delay difference between PWM0 and PWM1 block. In practice the delay value will compensate this.

4.4 Inverting

Inverting the channel is accomplished by changing the set and clear value in the match registers of the PWM block. This is done in the functions PWM_doubleEdge_on and PWM_doubleEdge_off when the input parameter inverting output is true.

4.5 Phase sequencer

The Phase sequencer of the demo is made with the PWM match 0 interrupt. The match 0 counter is used for switching to the next phase. For real applications an external interrupt together with a position sensor can be used. The sequencer is made in Main and is scanning Match_counter and calling PWM_switch function for the next phase. The switching pattern of the channels for the bridge is programmed in the latter function.

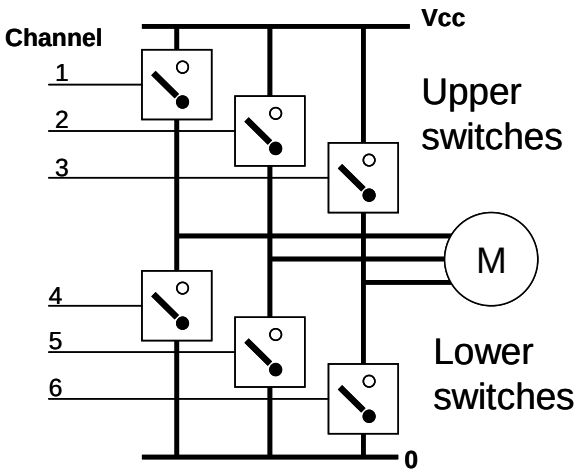


Fig 8

4.6 Channel switching pattern

The table shows that or an upper switch is changing or a lower switch. This pattern is programmed in the function PWM_switch. This is a common switch pattern for motor control applications.

CHANNEL \ Phase	1	2	3	4	5	6
1	on					on
2				on	on	
3		on	on			
4			on	on		
5	on	on				
6					on	on

Fig 6

5. The software

The software is developed on a Keil environment with Realview C compiler. The (Keil) startup file for the LPC23xx has been used. Also for the LPC2468 microcontroller which has been used for testing the final software. This file initialises besides the Stack, the Heap, the Mam also the clock setup. The Initialisation of PWM block, the cycle time, the duty cycle, the delay, the rise and fall time has to be adjusted for each application.

6. Conclusion

The PWM blocks of the LPC24xx allow controlling a bridge for (f.i.) motor control with switching dead time and with a minimum of Core execution load for the microcontroller. The (software) drivers can also be used for the LPC23xx (only 3 channels).

7. References

- Data sheet
- User manual LPC24xx/LPC23xx

8. History

Revision history

Rev	Date	Description
<00>	09032007	draft version

founded by

PHILIPS

© NXP B.V. 2007. All rights reserved.

Application information — Applications that are described herein for any of these products are for illustrative purposes only. Philips Semiconductors make no representation or warranty that such applications will be suitable for the specified use without further testing or modification.