Lego Car FPGA Demonstrator-Motor Controller

Dexin Chen
Technical University of Munich
dexin.chen@tum.de

History

05-12-2012 Version 1
20.02.2013 Version 1.1 change the user guide of servo motor
02.05.2013 Version 1.2 add description to component editor

Prospect

Design a PWM generator which conforms to the Altera Avalon specification. The IP core could be configured by software on Nios II to generate PWM with different duty cycles and frequencies.
Design a motor driving circuit to transfer the controlling signal from FPGA to motors and isolate the control unit with high voltage unit.

Principles

- DC motor

  The motor we use here is the LEGO Power Function Series, a detailed evaluation of this type of motor could be found in [1]
  As usual, a Pulse Width Modulation (PWM) could be used to control DC motor. By changing the duty cycle of PWM, the speed of a DC motor could be controlled.
  The frequency of PWM varies on different motors, empirically a frequency range from 10k to 30k is suitable, too low would possibly not make the motor function, too high frequency would lead to audible noise. The duty cycle should be able to vary from 0% to 100%.
  Power Function motor has a four wire connection to the control unit as Fig.1. For DC motor, C1 and C2 is used as power lines. By applying PWM to one of C1/C2 while set the other to 0, the motor will turn with the

speed according to the duty cycle of signals. Change the line which signal is applied will change the direction

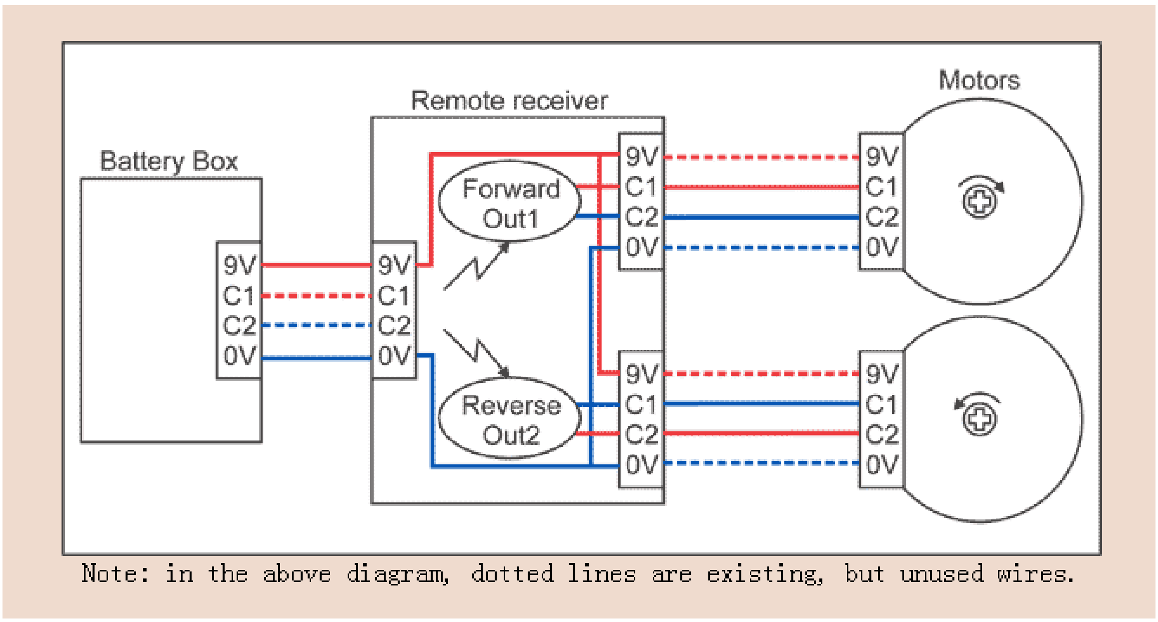

Battery Box
Remote receiver
Motors
9V
C1
C2
0V
Forward
Out1
Reverse
Out2
Note: in the above diagram, dotted lines are existing, but unused wires.

Figure 1

- Servo motor

  Servo motor we use is also Power Function series Servo motor. A detailed analysis of this motor could be found in [2].

  The controlling signal is also pulse width modulation (PWM) but in which the width of on pulse determines the position (angle) of the motor shaft and the amount of increment or decrement in the on pulse width determines the speed of rotation. The signal must be sent every 20ms (50Hz) and the range of duration for the motor is 0.5ms (0 degree) to 2.5ms (180 degree) [3].
  The LEGO Powerfunction servo motor is different. Its accepted frequency of PWM is around 10 KHZ.

- Avalon Bus

  The control unit we use here is Nios II. We customized a PWM IP module in SOPC builder and design the interface which complies with Avalon Specification [4]

- Motor driving IC

The driving IC we choose is L298N. Its total DC current is up to 4A and operation supply voltage is up to 46V. It could support up to four channels output.

The schematic of driving circuit is as Fig.2

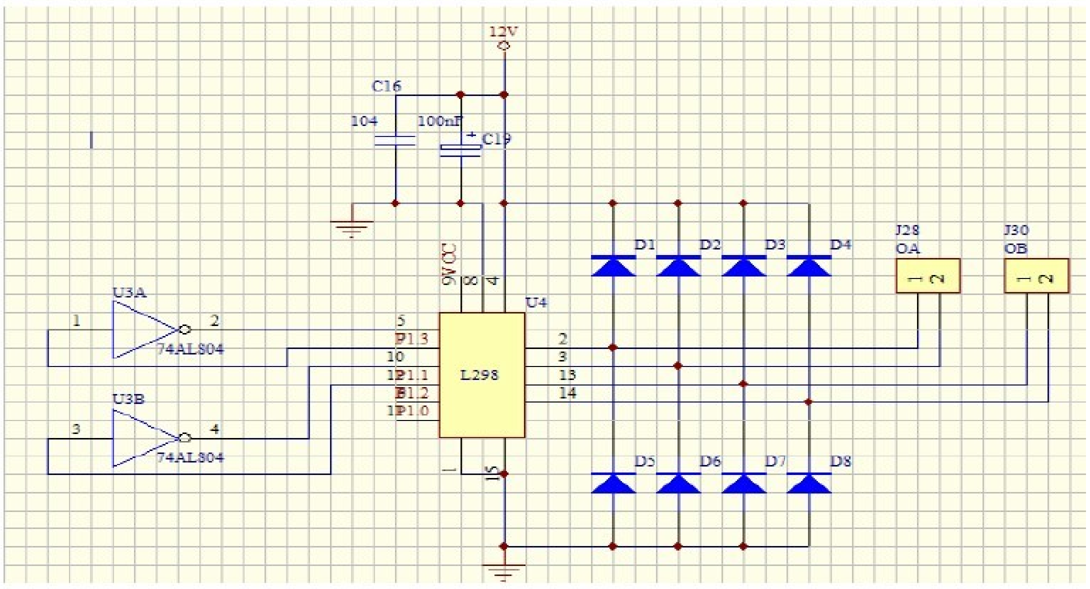


Figure 2

Application Guide

- Source code list

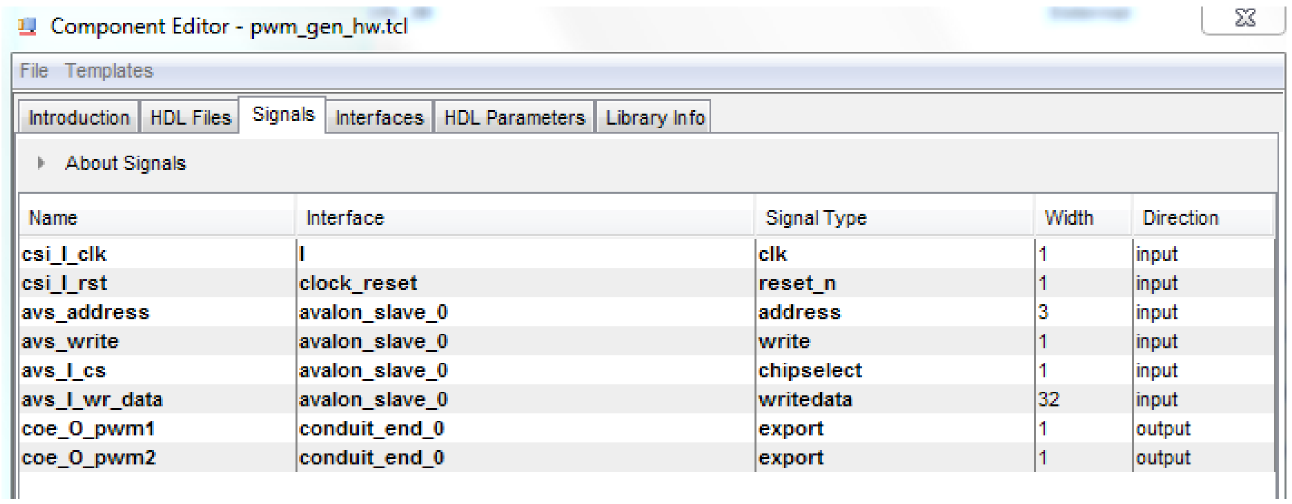
1. pwm_gen.vhd : vhdl module which implements the function of pwm generation. Its interface to Nios II is Avalon compatible.
2. pwm_gen_tb.vhd: testbench of pwm_gen.vhd file
3. motor_setting.h: header file of motor_setting.c, including some macros
4. motor_setting.c: it provides motor setting function to the user. By calling the function of motor_setting(), parameters of PWM such as phase shift, duty cycle, period and enable signal could be configured.
   Arguments of function motor_setting() is introduced here:

```
 *   phase: range from 0 to value of period
 *   duty cycle: set to number=period*percentage of duty cycle
 *   period: according to the freq of cpu,
 *       normally the period should be set to the value
 *       that makes the freq of pwm waveform to be 15 k
 *       e.x for 50MHz, value should be 3333(0xD05)
```

*   enable:'0' represents off, '1' is on, lease significant bit is for channel 1, second bit is for channel 2, e.x for channel 1 on and channel 2 off, enable=0x1

Steps

1. Instantiate a customized IP module in SOPC builder with pwm_gen.vhd using component editor. The signal tab in Component Editor is as Figure 3.



Name	Interface	Signal Type	Width	Direction
csi_I_clk	I	clk	1	input
csi_I_rst	clock_reset	reset_n	1	input
avs_address	avalon_slave_0	address	3	input
avs_write	avalon_slave_0	write	1	input
avs_I_cs	avalon_slave_0	chipselect	1	input
avs_I_wr_data	avalon_slave_0	writedata	32	input
coe_O_pwm1	conduit_end_0	export	1	output
coe_O_pwm2	conduit_end_0	export	1	output

Figure 3

2. After generate the BSP in Nios II, import the motor_setting.h and motor_setting.c file into the project.
3. Call function motor_setting() wherever you want and configure it.
4. For Lego servo motor the frequency is set to 13KHZ, duty cycle from 0% to 100%, the same for DC motor.
5. Only 2 pwm output per PWM components, so each time more pwm components are to be used, new components should be added in the SOPC builder. And copy the motor_setting.h and motor_setting.c file once again, give the function a new name e.g motor_setting_servo(). Change the base address accordingly, then call motor_setting_servo() anywhere you want.

Reference

[1] http://www.philohome.com/pf/pf.htm

[2] http://www.philohome.com/pfservo/pfservo.htm

[3] Position Control of DC Servo Motors Using Soft-Core Processor on FPGA to Move Robot Arm, O. A. Jasim, A. Z. Mansoor and M. R. Khalil, International Electrical Engineering Journal (IEEJ), Vol.2, No.3, pp.555-559, ISSN 2078-2365, 2011.
[4] Avalon Interface Specification Datasheet