Kris M. Vallozo
BSCpE- 3A

DC MOTOR INTERFACE

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity DC_MOTOR_INTERFACE     is
Port (
   clk      : in  STD_LOGIC;
   reset_n  : in  STD_LOGIC;
   start_n  : in  STD_LOGIC;
   dir_n    : in  STD_LOGIC;

   pwm_n    : out STD_LOGIC;
   motor_n  : out STD_LOGIC_VECTOR(1 downto 0);
      stat_led_n : out STD_LOGIC
);
end DC_MOTOR_INTERFACE;

architecture Behavioral of DC_MOTOR_INTERFACE is
   signal pwm_cnt    : unsigned(7 downto 0) := (others => '0');
   signal clk_div    : unsigned(18 downto 0) := (others => '0');
   signal pwm_clk    : STD_LOGIC := '0';
   signal enabled    : STD_LOGIC := '0';
   signal direction  : STD_LOGIC := '0';
begin

     process(clk)
   begin
      if rising_edge(clk) then
         clk_div <= clk_div + 1;
         pwm_clk <= clk_div(18);
      end if;
   end process;

   process(pwm_clk, reset_n)
   begin
      if reset_n = '0' then
         pwm_cnt <= (others => '0');
         pwm_n <= '1';
      elsif rising_edge(pwm_clk) then
         pwm_cnt <= pwm_cnt + 1;
         if enabled = '1' and pwm_cnt < 128 then
            pwm_n <= '0';
         else
            pwm_n <= '1';
```

```
            end if;
        end if;
    end process;

    process(clk, reset_n)
    begin
        if reset_n = '0' then
            motor_n <= "11";
            enabled <= '0';
            direction <= '0';
        elsif rising_edge(clk) then
            enabled <= not start_n;
            direction <= not dir_n;

            if enabled = '1' then
                motor_n <= not (direction & not direction);
            else
                motor_n <= "11";
            end if;
        end if;
    end process;

    stat_led_n <= not enabled;

end Behavioral;
```