

CECS 347 Spring 2018 Project #1

by

Doan Peter, Takatsu Naoaki

22, February 2018

A simple DC motor car controlled by PWM. 3 colors of the LED indicate the direction and movement of the car. The onboard switch 1 controls the speed of the car, while switch 2 controls the direction.

**Introduction**

In this project, students will create a simple DC motor car. The motor’s rotation speed will depend solely on PWM period-to-duty-cycle ratio. The onboard switch 1 will change the speed of the car to the following: 25%, 50%, 100%, 0% out of full speed. Switch 2 will toggle the direction of the car’s movement, either forward or backwards. The 3 LEDs will indicate the following: red light for stop, green light for forward, and blue light for backward.

**Operation**

Using the TM4C123, there are two on board switches used to dictate the direction and speed the car will use. SW1 is used for the speed, which when pressed, it will cycle through 4 speed cycles. The first, which is the initial, is no speed. The car will not move. Then it will go to 25%, 50% and lastly 100%. SW2 is used for the direction, which when pressed, it will go the opposite direction it is currently going. Initially, the car is on a forward setting, so when it is pressed for the first time, it will go backwards. Furthermore, there are 3 LED’s on the board are used to represent what action is currently being committed. When the LED is red, then the car is not moving. When blue, it is moving backwards. When green, it is moving forward.

Going into further detail, when SW2 is pressed, Port A2-5 sends signals based on the forward flag. When forward flag is 1, then it will send a 1 to PA2 and PA4 while PA3 and PA5 gets a 0. Doing so will result in the car moving forward. Likewise, when forward flag is 0, then it will send a 1 to PA3 and PA5 while PA2 and PA4 gets a 0. This results in moving backwards.

SW1 controls the PWM that is sent out from Port B6-7. The duty cycle is multiplied based on which setting it enters, either a .25 for 25%, .5 for 50%, and the full period when at max motion.

<https://www.youtube.com/watch?v=4t5FictVMNg>

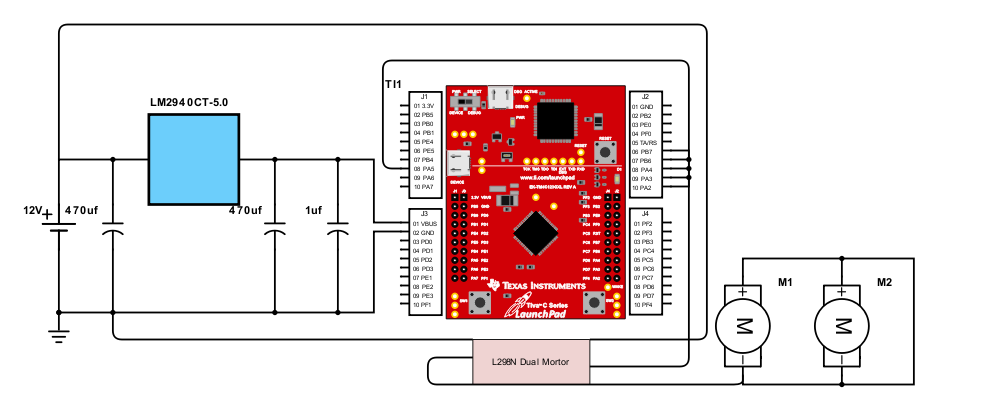
**Theory**

Motor: The motor’s rotation speed is controlled by PWM (Pulse Width Modulator) duty cycles. To obtain certain amount of speed, we set the duty cycle equal to the percentage of the speed we want multiply by the whole period. Since the duration of the duty cycle directly connects to the duration of how long the motor runs each period, we can set the % to be higher for faster speed and lower for slow speed.

LED: Indicates the movement/direction status of the car. We set a boolean flag called forward to check the direction and unsigned char called speed\_menu to check the car’s speed. Utilizing if else statements, we can make the LEDs toggle to the status we desire.

Switch: Switch 1 changes the speed of the car while switch 2 changes the direction of the car. Since we use the onboard switches, we needed to unlock port F in order to use these pins.

**Hardware Design**

****

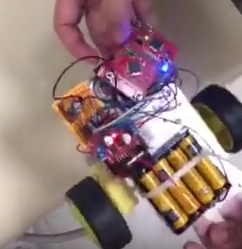
**Software Design**

The software includes 2 port initiations, for Port A and Port F. Port A initializes the signals controlling the wheels while Port F initializes the signals controlling the onboard LED’s and onboard push buttons. There is then two PWM initiations for Port B6 and Port B7 that controls the signals that controls the speed of the wheels.

When SW2 is pressed, the forward flag is set to the inverse of the current value. After it gets out of the if condition for SW2, it will then go through condition checks, first checking if the car is moving. If it is moving, then it will go to the next part of the if statement, else it will output the color red. If it is moving, then it will check the forward flag. If the flag is a 1, then it will set PA5-2 to 0x14 and then output the color green to the onboard LEDs. If the flag is a 0, then it will set PA5-2 to 0x28 and then output the color blue to the onboard LEDs.

When SW1 is pressed, it will increment the speed menu counter and based on the current value, it will change the duty cycle. When the speed menu counter reaches the value of 5, it will reset the value of itself to 1. When it is 1, the car will proceed at 25%. At 2, it will go at 50%. 3 will be 100% while 4 will be 0%, or not going at all.

All these signals are connected to its respective locations on the L298 Dual Motor controller. PA2-5 will be connected to IN1-4, PB6 will connect to ENA and PB7 to ENB. IN1-2 and ENA controls the first motor while ENB, IN3-4 controls the second motor. IN1-4 controls the direction, so when a signal is sent to IN1 and 3, then it will go forward. While IN2 and IN4 gets a signal, it will go backwards.



**Conclusion**

In this project, we have spent significant number of hours debugging the code. Firstly, the onboard LED did not light up as we have expected; later we found out that our if else condition had flaws. Next, our PWM powered motor did not work at all; by changing the pins from PA6,7 to PB6,7 and rebuilding the code from scratch, we have managed to make the motor running. Lastly, when the direction toggle button was pressed upon the case where the motor is stopped, the motor ran itself when it should have not done so; to solve this problem, we have added another condition in the if statement. As for our success, our hardware circuit connections on the car has been working since our first trial. However, the most successful achievement of this project would be the knowledge and experience gained from all of our failures and problem-solving skills.