CECS 347 Embedded Systems II

Project 1: Robot Car with Motor Control

By: John Yu and Dr. Min He

Project Description:

This project will help students practice how to use hardware PWM and PLL hardware components, and review the following topics: GPIO and interrupts. You are to build a wheeled robot that changes its speed and direction according to the input switches of the TM4C123G LaunchPad Microcontroller. You can build your own wheeled robot or follow the guide below for this project.

Project Requirements:

1. Switch 1 will be used to increase the speed of the robot (switch debounce is required)
   - Robot will start with no motion, press once will put the robot in starting speed, around 30% duty cycle, keep pressing the robot speed will go through 60%, 80%, 98%, stop and the cycle repeats. You are required to use **hardware PWM** for speed control.
2. Switch 2 will be used to control moving direction of the robot (switch debounce is required)
   - Robot will start in forward direction mode, press once the robot will be in backward direction mode, press again and the cycle repeats.
3. RED LED indicates no robot motion
4. BLUE LED indicates backward direction
5. GREEN LED indicates forward direction
6. Hardware PWM implementation: you are free to choose two hardware PWM outputs available on TM4C123 Launchpad to drive the two DC motors.
7. Choice of system clock: you are required to use **PLL** to generate 50MHz system clock.
8. Submit a Report

Reference Project: TExaSware/C12_DCMoter, HardwarePWM, SysTickPLL

Starter Project: Lab 2

Recommended development steps:

1. Start with Lab 2, choose your PWM outputs for two motors. Test speed control with the two push buttons on the Launchpad and two LEDs on breadboard. Modify function GPIOPortF_Handler() SW1 touch logic to cycle through stop, 30%, 60%, 80% and 98% speed. If you have Analog Discovery 2, test your PWM outputs with Analog Discovery 2.
2. Build your Robot Car: follow the description given in the next section.
3. Connect your PWM outputs to the two DC motors driver PWM signals, test speed control using onboard push button SW1.
4. Add direction control to your robot car: figure out which pins to use for direction control. Modify function GPIOPortF_Handler() SW2 touch logic to change direction of the motors.

5. Add LED indicator for speed and direction: Add GPIO initialization code to use the three on-board LEDs. Modify GPIOPortF_Handler() SW1, SW2 touch logic to take care of direction LED changes.

Sample Robot Build:

Robot kit off of amazon for $11.99 and an H-bridge driver for $8.99 from amazon. See links below for details
Emgreat® Motor Robot Car Chassis Kit with Speed Encoder wheels and Battery Box
H-Bridge driver or L298N Motor Drive Controller
Robot Kit comes with almost all the pieces you need.

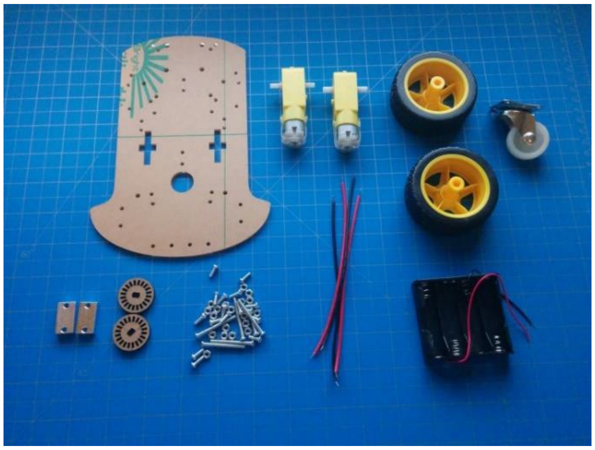


Figure 1: Robot Kit Parts

Insert the wires into the motor terminals and solder them. Careful with the terminals as they are very flimsy and when soldering do not overheat the terminals as this will melt the plastic housing of the motors.

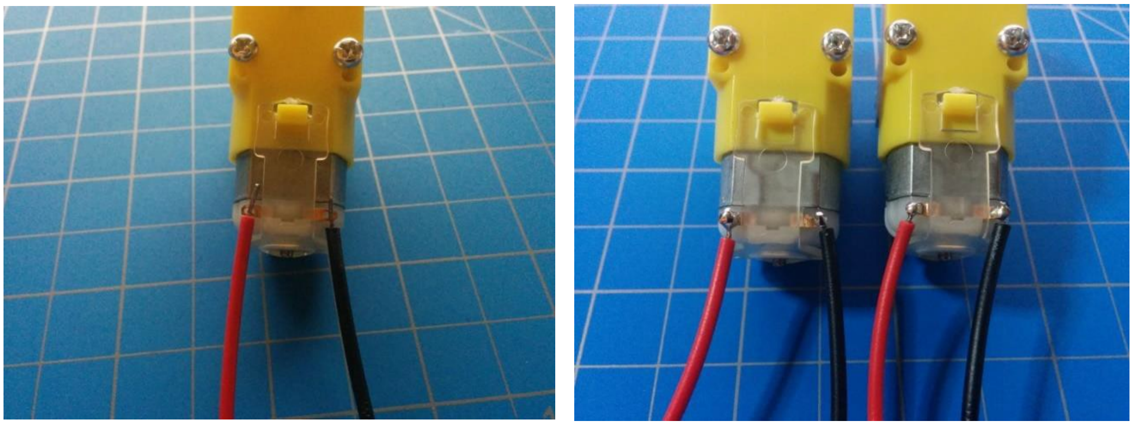


Figure 2: Wires Solder onto motor terminals

Note that DC motors do not have polarity so do not worry about flipping the wires.

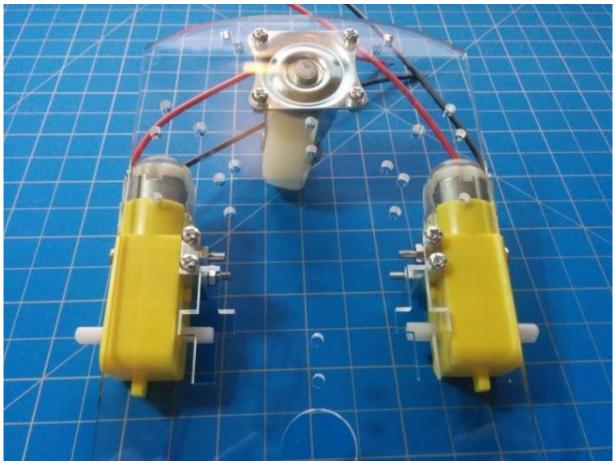


Figure 3: Motors and Caster wheel mounted

Mount the 2 DC motors and Caster wheels onto the acrylic platform.

Next we will put together all the circuitry to the robot. What you will need are the following:

1. The H Bridge driver L9110S, 5V LDO Regulator (LM2940CT-5.0) and two 470uf one 1uf capacitors to power board. Or one L298N motor driver.
2. TI Launch Pad (TM4C123) Microcontroller
3. 8xAA Batteries
4. Wires

Instructions on using L9110S motor driver:

The pin connection shown below is an example. You are free to choose your own pins for PWM outputs and direction control.

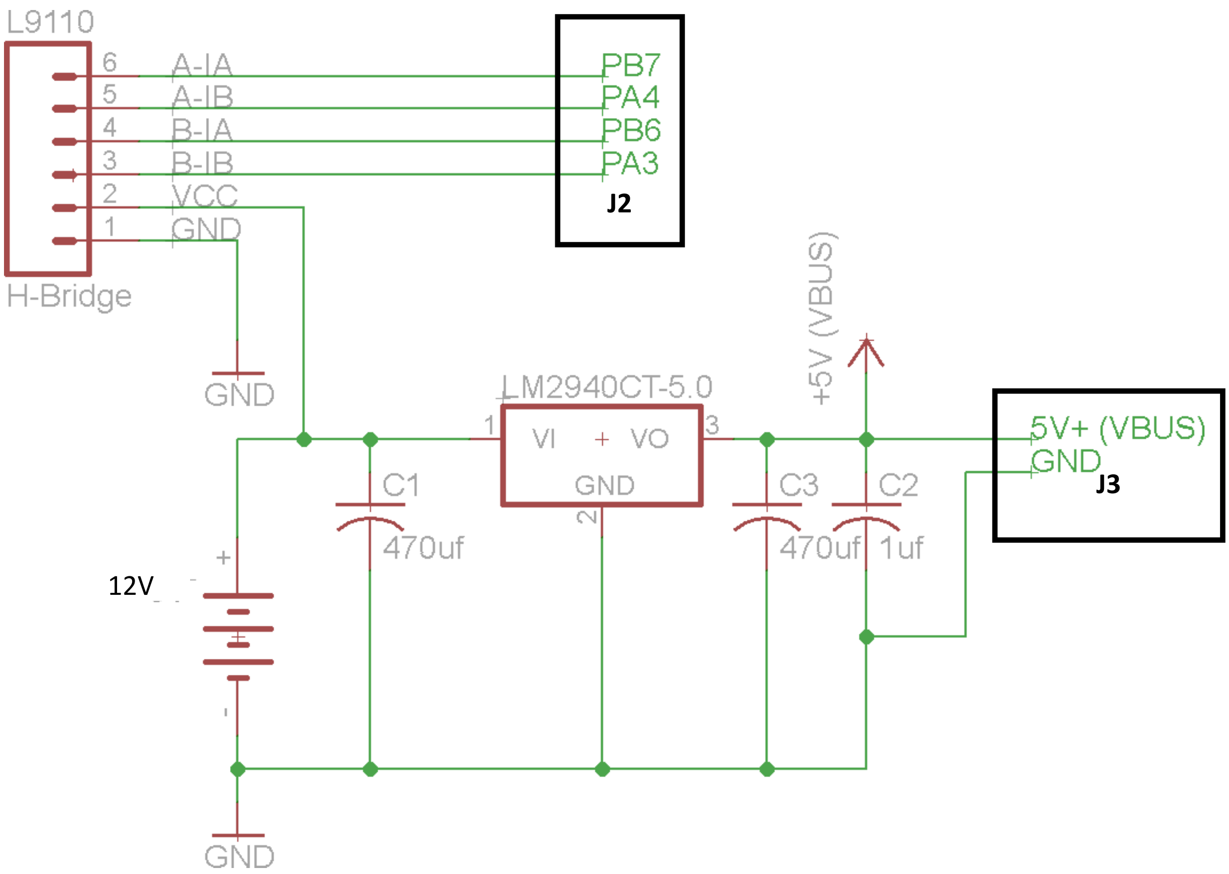


Figure 4: Schematic of the circuitry

TM4C123 Microcontroller requires either a USB power connection or 5V to the VBUS. Since the robot needs to be untethered from the computer a USB cable is not an option. Therefore, a LM2940 5V LDO regulator is used to regulate the power for the microcontroller. (Do not feed more than 5V into the VBUS, doing so will damage the TM4C123)

The H-Bridge has 4 control pins that can be used to control 1 stepper motor or 2 DC motors. In this application 2 DC motors will be used (Motor A and Motor B). The schematic shows that it's connected to J2 but feel free to connect them to any appropriate GPIO. The microcontroller will control the H-Bridge which will in turn drive the motors. Below is a table to help aid in programming the controls.

H Bridge Pins	Tm4c123 Signal
Motor A-IA	PWM (Speed)
Motor A-IB	DIRECTION (High or Low)
Motor B-IA	PWM (Speed)
Motor B-IB	DIRECTION (High or Low)

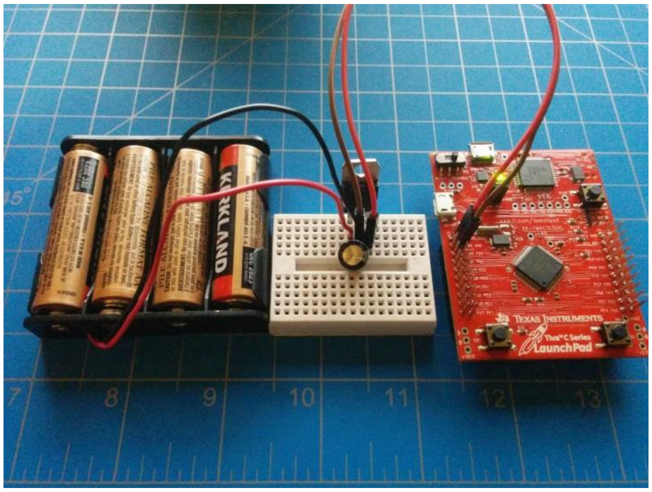


Figure 5: Testing the 5V Regulator

Hook up the 5V LDO regulator to the breadboard and attach it to the battery. Use a multimeter to test the voltage before hooking it up to the tm4c123. If the voltage is above 5V the tm4c123 will be damaged. Feel free to solder the regulator onto a circuit board or leave it on the breadboard for quick prototyping.

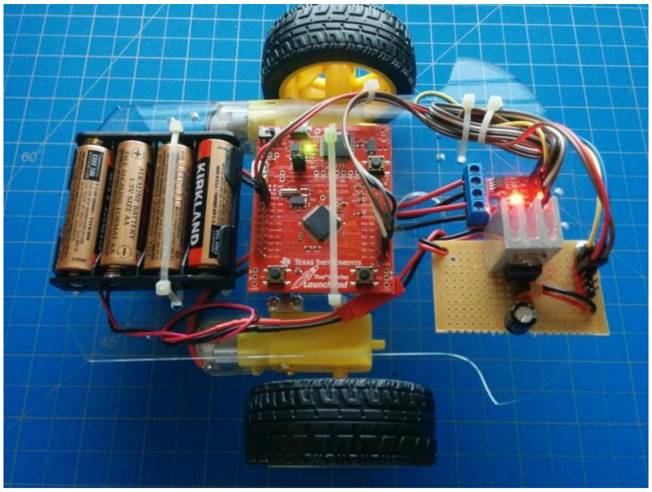


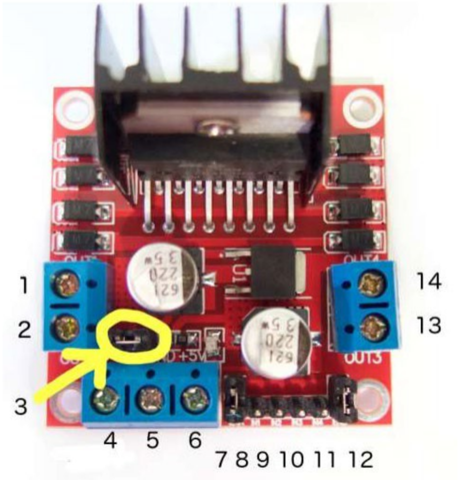
Figure 6: Assembled robot

Attach everything to the platform and program your robot. Be creative with your robot layout, the robot does not have to look like the above image. A good tip is to elevate your robot wheels and watch the speed and direction of the wheels while you are programming it. Have Fun and be Safe.

<u>Instructions on How to Use L298N Motor Driver:</u>

L298N H-bridge Dual Motor Controller Module allows you to control the speed and direction of two DC motors. The L298N H-bridge module can be used with motors that have a voltage of between 5 and 35V DC. There is also an onboard 5V regulator, so if your supply voltage is up to 12V you can also source 5V from the board.

First we'll run through the connections, then explain how to control DC motors.



1. DC motor 1 "+" or stepper motor A+
2. DC motor 1 "-" or stepper motor A
3. 12V jumper - remove this if using a supply voltage greater than 12V DC. This enables power to the onboard 5V regulator
4. Connect your motor supply voltage here, maximum of 35V DC. Remove 12V jumper if >12V DC
5. GND
6. 5V output if 12V jumper in place, ideal for powering your Arduino (etc)
7. DC motor 1 enable jumper. Leave this in place when using a stepper motor. Connect to PWM output for DC motor speed control.
8. IN1
9. IN2
10. IN3
11. IN4
12. DC motor 2 enable jumper. Leave this in place when using a stepper motor. Connect to PWM output for DC motor speed control.
13. DC motor 2 "+" or stepper motor B+
14. DC motor 2 "-" or stepper motor B-

To control one or two DC motors: First, connect each motor to the A and B connections on the L298N module. If you're using two motors for a robot (etc) ensure that the polarity of the motors is the same on both inputs. Otherwise you may need to swap them over when you set both motors to forward and one goes backwards! Next, connect your power supply - the positive to pin 4 on the module and negative/GND to pin 5. If you supply is up to 12V you can leave in the 12V jumper (point 3 in the image above) and 5V will be available from pin 6 on the module. This can be fed to your LaunchPad vbus pin to power it from the motors' power supply. Don't forget to connect LaunchPad GND to pin 5 on the module as well to complete the circuit.

Two PWM signals connected to pin 7 and 12 respectively will be used to control the speed of the two DC motors. Four digital signal signals connected to In1 – In 4 will be used to control the direction of the two DC motors in the following way: each motor needs two digital signals, for example for motor one, send a

HIGH to IN1 and a LOW to IN2 will cause it to turn in one direction, and a LOW and HIGH will cause it to turn in the other direction.