California State University, Long Beach

Cecs 347

Project 1

Rodrigo Becerril Ferreyra
Student ID 017584071

A project that uses the ARM Cortex-M4 microcontroller to drive a car.

05 March 2021

1 Introduction

The purpose of this project is to combine many different concepts learned throughout the course so far: hardware PWM, PLL, and DC motors. The requirement was to build a robot car that could move at various different speeds, and move forwards and backwards. My specific car has four wheels and four motors, so I required double the amount of batteries.

2 Operation

My program operates as follows: first, all the required clocks and registers are initialized once. This sets the microcontroller up to its default running conditions (in this case, forward operation at 30% speed, with a green light). After initialization, nothing else happens until an interrupt is detected (the two on-board pushbuttons both trigger an interrupt). If the left button is pressed, the speed of the car cycles as follows:

$$30\% \rightarrow 60\% \rightarrow 80\% \rightarrow 98\% \rightarrow 0\%.$$

The speed moves to the right every time the left button is pressed (0% loops back to 30%). When the right button is pressed, the car switches directions.

Here is a link to the operation of the car: https://youtu.be/z8ZxZ9zTwM4.

3 Theory

The microcontroller uses PWM (pulse-width modulation) to control the speed of the car. PWM is a technique in where a signal is repeatedly switched on and off to control the average power delivered to a load. In this case, the load is a motor: if the driving PWM signal is using a 30% duty cycle, the motor will only spin at 30% its angular speed.

This project also uses PLL (phase-locked loop). The PLL module in the microcontroller allows the system clock of 16 MHz to be increased up to 80 MHz. For this project, I am using a clock speed of 50 MHz.

4 Hardware Design

Figure 1 is a schematic diagram of my project. As noted in my video, I am using 8 AA batteries, making my power source 12 V. Figure 2 is an image of the physical car.

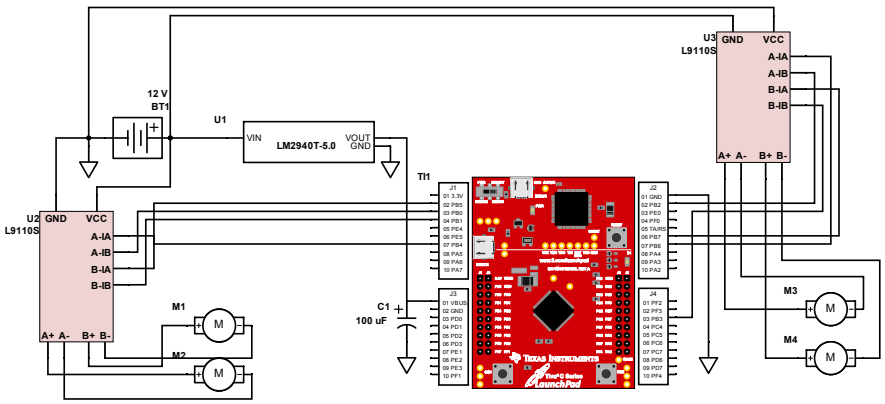

Figure 1: Schematic Diagram of system.

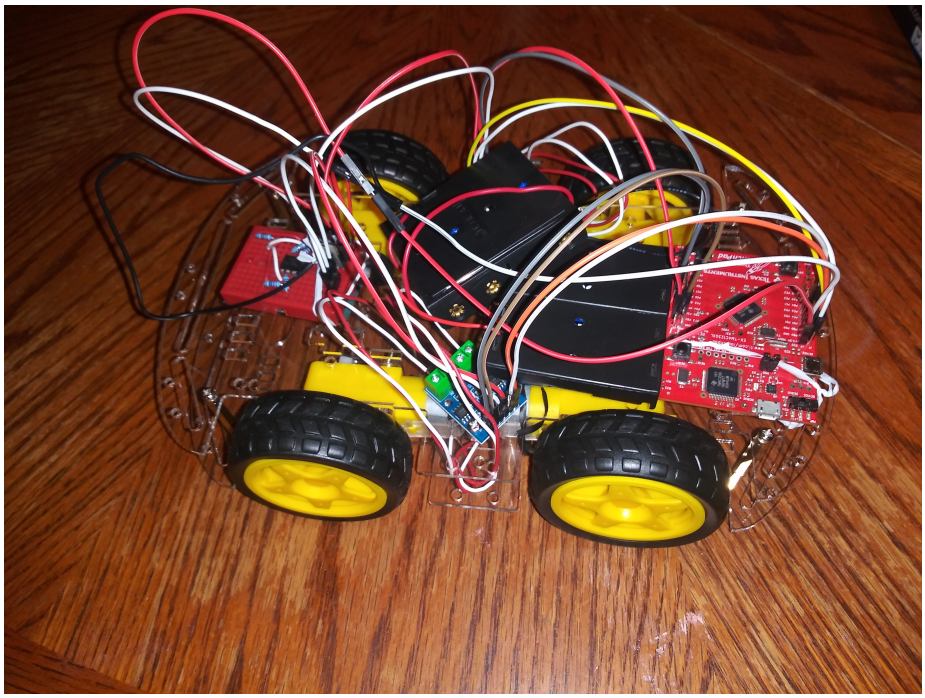

Figure 2: Picture of system.

5 Software Design

I decided to put all the logic in the interrupt handler, to make things simple. Originally, I was having issues, because when the car was in the 30% setting in the forward direction, if the user switched the direction, the wheels would spin at 70% speed. I managed to fix this, however.

6 Conclusion

Overall, I learned a lot from this project. I learned how many different elements (such as PLL and PWM) can come together to create one coherent system. I also learned the particulars in setting up each element, what registers are involved, and what can and cannot be done and in what order.