

## L293D-Project README

### Project Location

The final project can be viewed [here](#).

### Project Description

This project involved designing a system to issue mobility commands to a rover. For the purposes of this assignment, this "rovers" mobility was expressed as a system of two motors, one on each side. The designed system allows for the user to move the rover forwards and backward and to turn the rover left and right. These actions are achieved by toggling one of the five switches on the board. The user can also modify the speed of the rover by turning a potentiometer.

### Project Details

Some deviations were made from the proposed design, as can be seen in *Figure 1*.

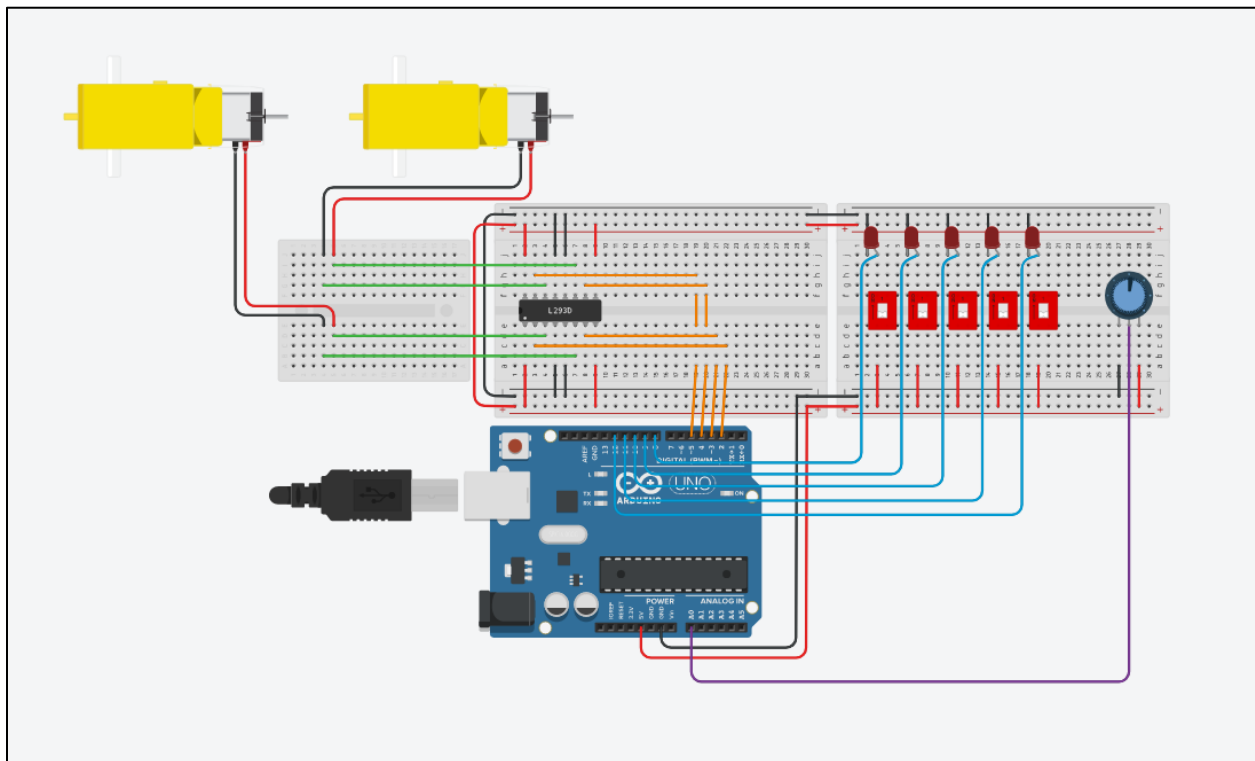


Figure 1: Final System Design

Due to technical problems, five individual switches were used as opposed to the single six-switch dip switch. When attempting to use the recommended switch, the values were not being correctly read into the Arduino, which was likely a result of a wiring error. This was fixed by adding individual switches with an LED which was mainly used to debug the system. The LEDs were kept in for user convenience.

Additionally, three boards were used instead of one. This was mainly done to try and maintain some sort of orderliness in the wiring. As shown, the final result is "passible" in this respect. Lastly, the most noticeable difference is the lack of an integrated potentiometer. Again, due to technical issues (these of which I cannot explain to save my life) this feature was cut, essentially leaving the user "brakeless" (with the exceptions of the reset switch). For some reason, the analog commands (particularly *analogueWrite()*) caused unusual behavior in the motors which caused them to malfunction. For example, both motors would refuse to spin in the positive direction and wouldn't fully spin in the negative direction. Due to this, all motors now rely on digital commands, which solved the abnormal behavior. As a consequence of this, the speed of the motor was unable to be scaled by the potentiometer, although the values for it are still gathered and stored by the system.

### **User Instructions**

To use run this project, click Start Simulation at the top right of the screen. Once the project is running, you will have access to the switches and potentiometer. Throughout the duration of the run, the values of the switches (represented as 5 binary values), potentiometer, and "rotation speed" (which isn't accurate) will be shown in the Serial Monitor. To issue a command to the rover, simply turn on a switch. The descriptions for what each switch does can be found in the source code. For simplicity, they will be restated below:

**COMMAND DESCRIPTIONS**

Switch 1 = Move Forward  
Switch 2 = Move Backwards  
Switch 3 = Move Right  
Switch 4 = Move Left  
Switch 5 = Reset

It should be noted that only one command can be active at a time, due to the design of the system. The first-toggled switch will be preferred.

The potentiometer can be modified by rotating its dial. The corresponding results can be viewed in the Serial Monitor.

**Final Progress**

The final state of the project is complete, except for the potentiometer (likely due to my own idiocy). The user can move the rover forwards and backward and turn the rover left and right.