RC Car Project

Team Speed Racer

Michael F. Seale (Team Leader)

Teammates Names Redacted

Table of Contents

Introduction

An RC car without its remote may seem like garbage to most. The field of engineering, however, provides practical solutions to such problems. The aim of the project is to control an RC car over wifi using a mobile app. Utilizing the RemoteXY app to interface with the model vehicle enables the user to use their own familiar hardware, and additionally presents opportunities to customize the features and software interface to meet one's own needs.The choice to communicate with the car over wifi rather than traditional radio signals reflects the advancement of wireless technologies and the increasing demand for everyday systems to be controlled remotely over internet protocols.

Design Process

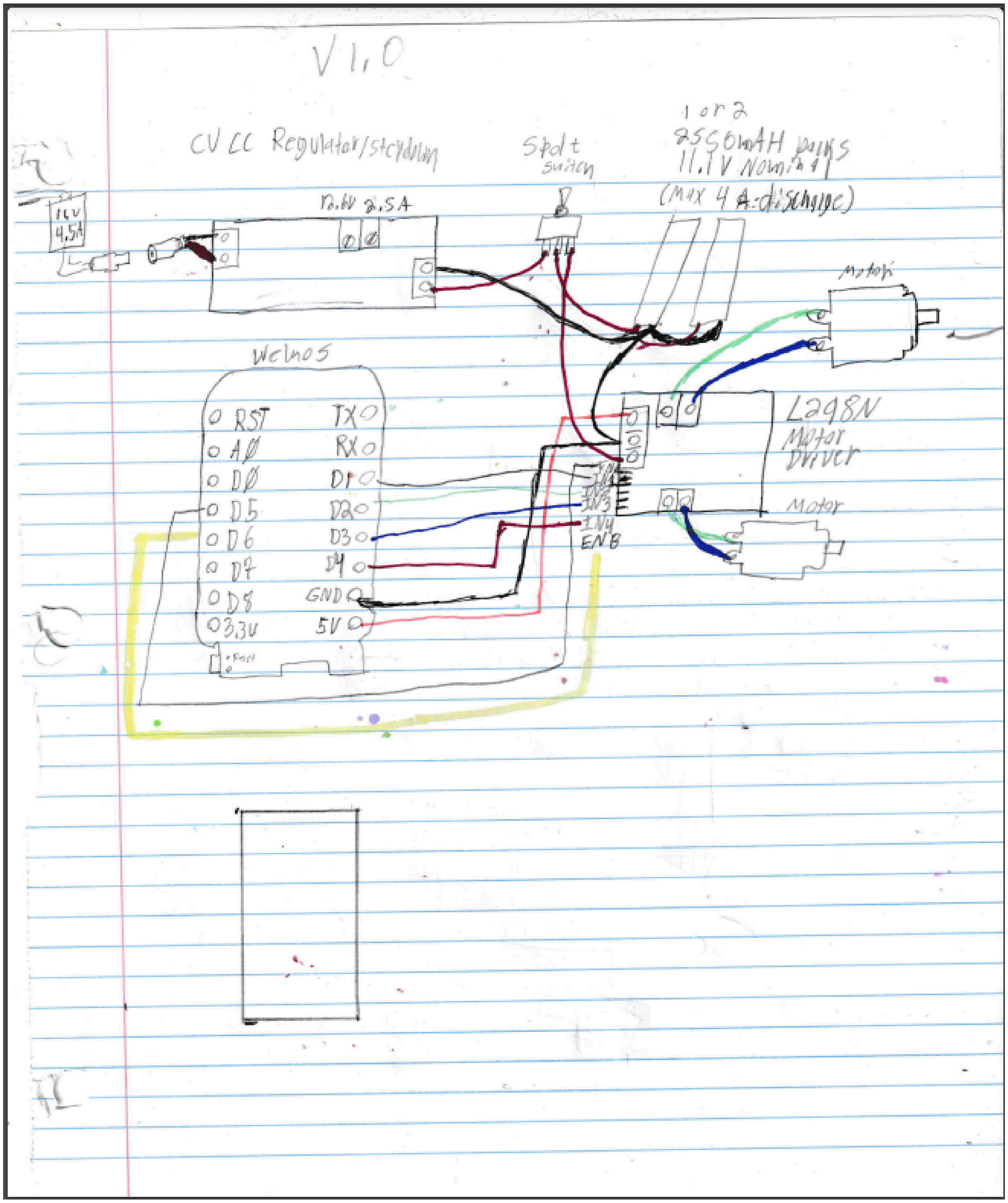
Our team chose to meet in the engineering lab during the scheduled class time on Wednesdays in order to discuss our plans and work on the project together.The general approach of using a microcontroller and a dc motor controller in order to control the vehicle's movements was discussed in class, whereas the specifics of the design relied on the technical specifications of the vehicle we chose to modify as well as the specific features we chose to implement.

We initially began by disassembling and rebuilding our RC vehicle in order to gather an idea of the components it consisted of and how they worked together. With this information in mind, we continued by brainstorming which sort of additional features we'd like to add to the basic concept. In doing so, we agreed we'd implement headlights and, if time allowed, sound functionality. We decided to continue onward with our project using the evolutionary prototyping method.

Version 1.0

Version 1.0 of our RC car project represented our attempts to produce basic functionality in interfacing with the model vehicle over the RemoteXY app. We began by developing an electronics schematic based on the specifications of our car and the information presented in class. We continued by wiring the basic circuit and testing its functionality. Finally, we designed a simple interface on RemoteXY and combined the generated code with the code provided to the class in order to test the system before mounting into the vehicle. While working on this iteration, we destroyed our first microcontroller by powering it on 12v instead of 5v.

v1.0 Electrical Schematic



Experimental Procedure

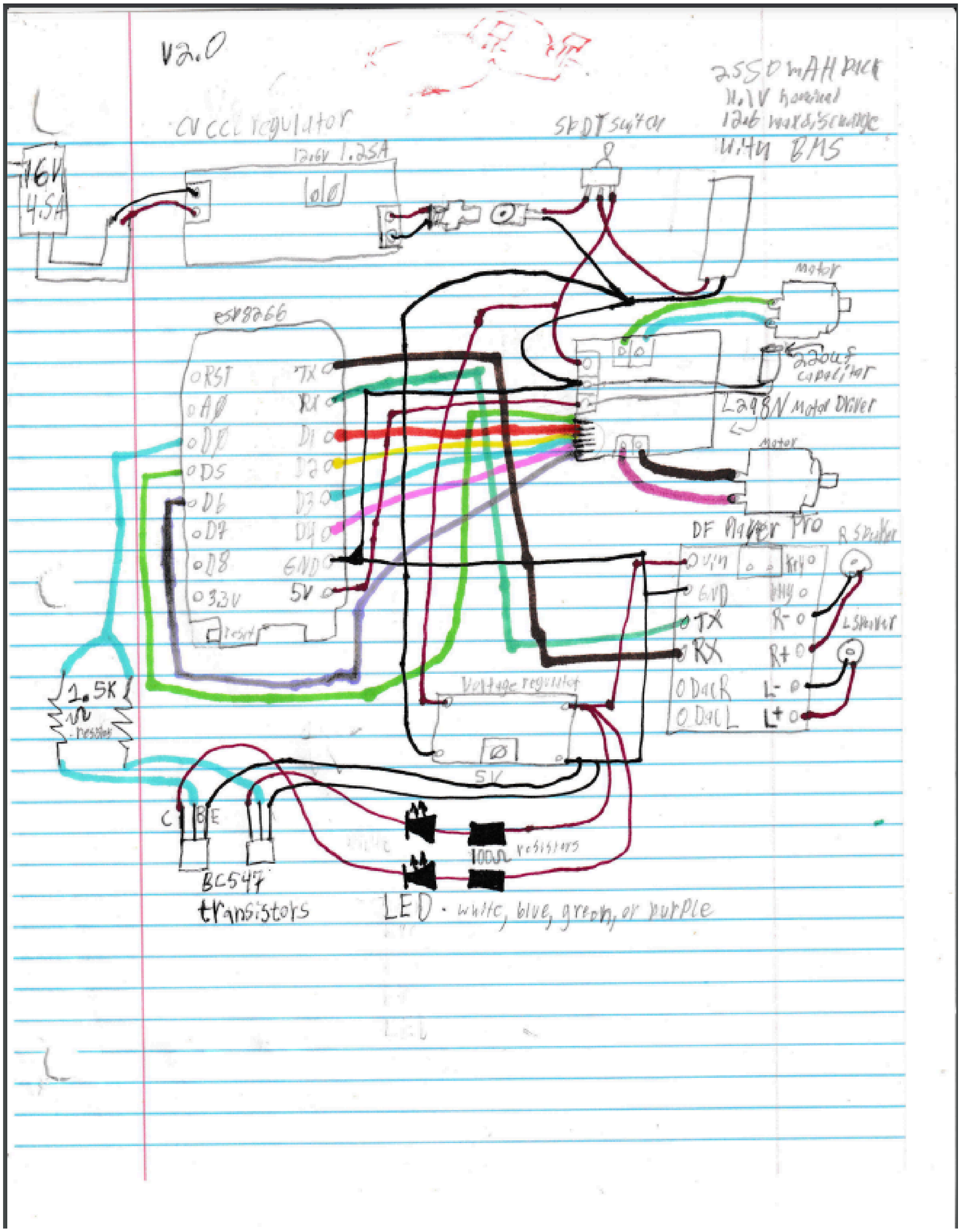
We first tested the functionality of our initial design by connecting our basic circuit to the vehicle’s motors and testing them over RemoteXY. We verified that we were able to drive forwards, drive backwards, and turn left. However, we were not able to turn the vehicle right. After mounting the components, we tested the vehicle’s mobility. It was then that we noticed connectivity issues. By driving the vehicle over small obstacles and into obstructions, we were able to verify its otherwise sufficient mobility, power, and durability.

Continual Revisions

Version 2.0

The second iteration of our design focused on responding to the problems we discovered with the first version as well as implementing headlight and sound functionality. The code was revised so that pin 2 was properly initialized, enabling right turns. The pwm values were also adjusted for the steering motor in order to allow for a full range of motion. In order to resolve wifi connectivity issues, a capacitor was placed on the 5v output of the motor controller, to provide additional current. As only one GPIO pin (without additional duties) was free, we used transistors to act like a switch in order to control LED headlights. We created an electronics schematic to demonstrate our plans for these additions as well as our plans to implement audio. Speakers were planned to be driven by an mp3 module which was controlled over serial communication. A 5v regulator was to be used to power LEDS and sound.

v2.0 Electrical Schematic



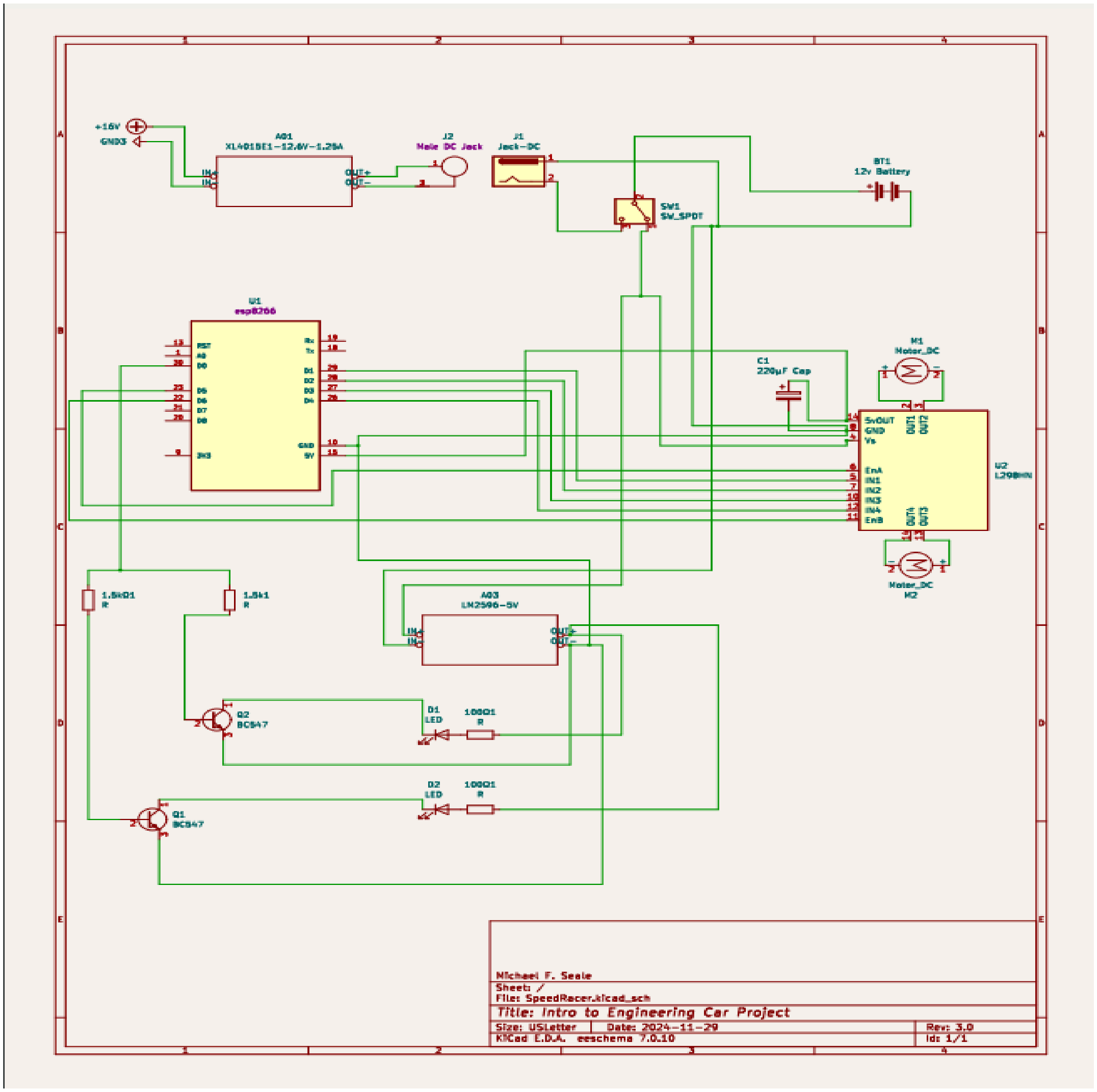
Continual Experimentation

Our headlights, vehicle mobility, and software controls were verified to be functioning as expected. The range of the vehicle was tested in the ground floor of Goodwyn hall, to satisfactory results. However, The serial communication for the sound was found to be interfering with wifi, preventing the vehicle from connecting with the RemoteXY app. Despite testing various configurations, the sound was consistently found to interfere with wifi communications.

Version 3.0

For version 3.0, various physical aspects of the device were improved to include proper mounting of components and improved wiring. Headlights were mounted to the outer shell of the vehicle. However, at this point, sound was dropped from the design. After testing using serial commands directly, two microcontrollers communicating with i2c, and two microcontrollers communicating by digital gpio pin outputs, all configurations were found to interfere with wifi. As the focus of the project is on controlling the device over wifi, audio capabilities were dropped from this final design. A final electronics schematic was drawn in KiCad. A wooden chassis was constructed for the charger.

v3.0 Electrical Schematic



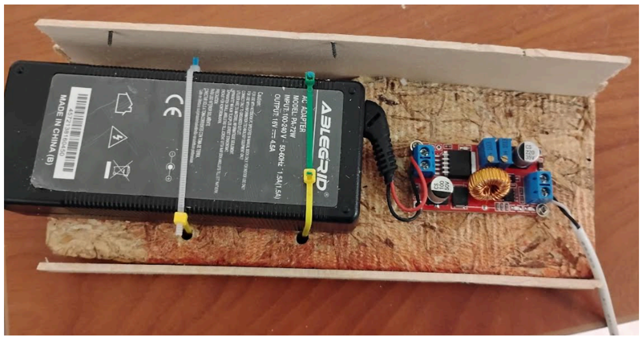
Technical Information

Battery, Switch, and Charger

An 11.1V nominal voltage, 2.55 amp hour lithium ion battery pack was mounted at the top of the roll cage using velcro straps. This battery pack consists of 3 EVE 18650 cells arranged in a 1p3s configuration with an integrated BMS. It was originally recovered from powered window blinds. The max voltage is 12.6v and the max discharge rate is 2 amps. We chose this battery because of its high quality cells and because it's within the safe operating parameters for our vehicle's motors.

The common terminal of a single-pole, double-throw switch was wired to the positive side of the battery. One of the throw terminals was connected to the various components of the device while the other throw terminal was connected to the DC jack for charging.

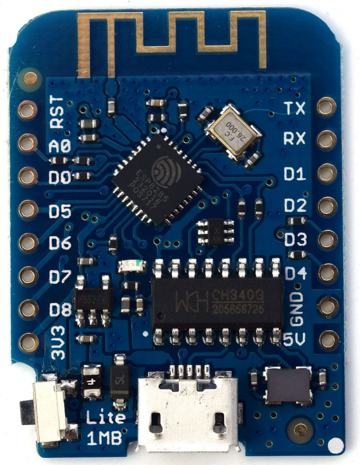
A 16v laptop charger was repurposed in order to provide DC power to the battery. Its output was wired into the input of a constant current, constant voltage regulator in order to charge the battery safely. The regulator was set to 12.6v and 1.25 amps, according to the battery cell's charging curve available in its documentation. The regulator's output was connected to a dc barrel jack.



Microcontroller

The Wemos d1 mini lite ESP8266 module was chosen based on the instructor’s recommendation and its ability to operate on 5 volt input. The 5V pin is connected to the 5v output of the motor driver. GND is connected to ground on the motor driver, which is connected to the negative terminal of the battery. This GND wire is also spliced to the negative side of the output of the voltage regulator that powers the LEDs and sound, in order to allow for communication with all devices that are controlled by the ESP8266.

Pins D1 – D6 are connected to the inputs of the microcontroller. Pin d0 is used to control the LED headlights. Pins D7, and D8 were avoided because they have other functions on the microcontroller that we didn’t want to risk interrupting. Pin A0 was unused because it is meant for analog inputs.

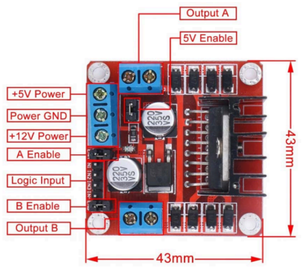


Motor Driver

Our motor driver module is based on the L298N IC. It was chosen based on the professor’s recommendations and because the code provided was written with this module in mind. From the module, 12V is connected to the switch. Ground is connected to the negative terminal of the battery, as well as GND on the microcontroller in order to enable communications. 5V out powers the microcontroller. Data pins ENA, ENB, and IN1 – IN4 were connected to GPIO pins on the microcontroller, according to its documentation. Outputs A and B were connected to the steering motor and driving motor.

A 220 microFarad electrolytic capacitor was placed between ground and +5v out. This was used to ensure that the ESP8266 does not draw more current than it is available when using wifi.This prevents wifi from cutting out when current spikes.

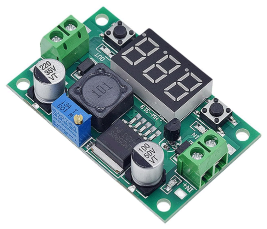
: A regular DC motor turns the back wheels of the car. The PWM values for this motor were adjusted in the code to 200, so the motor turns about 80% as fast as possible. Our steering mechanism uses a regular DC motor and a gearbox rather than a servo. Due to the spring present in the gearbox that keeps the wheels aligned straight when not turning, PWM values for the steering motor had to be turned up to 255 rather than 100, so that the wheels would turn all the way.



Voltage Regulator and Headlights

We used a module based on the LM2596 IC to drop the voltage to 5 volts in order to power the LEDs and the sound. The 5v output of the motor driver was not used for this purpose in order to prevent overcurrent draw. The negative output of the voltage regulator is connected to GND on the microcontroller in order to allow for communication with the devices it powers. This regulator was chosen because of its availability, price, and voltage range.

Each 20mA LED is wired in parallel. The positive output of the voltage regulator is connected to a 100 ohm resistor, which is connected to the anode of the LED. Two BC547 NPN transistors are used as a switch to control each LED. The cathode of the LED runs to the collector of the transistor. The emitter of the transistor is connected to the negative output of the voltage regulator. The base terminal of the transistor is connected to a 1.5k ohm base resistor in order to provide the proper control current to the transistor. Each of these base resistors are connected to the D0 pin of the microcontroller. When the output of this pin is low, the transistors will act as an open switch and the LED will be off. When the output of the pin is high, the transistor will act as a closed switch, turning on the LED. These particular transistors were chosen due to availability and appropriate electrical characteristics.



Costs Breakdown

Item	Cost ($)	Cost of those used ($)
Breadboard Wires	2	2
Motor Controller	5	10
ESP8266	5 (x2)	10
Red and black electrical wire	11	6
CCCV regulator	6	6
Switch	10	5
Velcro	3	3
Voltage Regulator	6	3
Capacitor	4	0.02
Resistors	5	0.04
LED’s	10	0.08
Transistors	13.84	0.05
RC Car	10	10
Heatshrink Tubing	7	0.75
Battery	5	2.50
Total ($):		56.44

Sources

Allecin, “24 Values Multicolor LED Light Emitting Diodes Kit,” Amazon, https://a.co/d/hzFHib6

EVE Energy, “Lithium-ion rechargeable cell,” RD-EVE ICR18650/26V-S118-LF Datasheet, Dec. 2020

Fairchild Semiconductor, “NPN Epitaxial Silicon Transistor,” BC546/547/548/549/550 Datasheet, Aug., 2002

Oyito, “DC-DC 5A Adjustable Buck Regulator Power Supply Module User Manual,” https://1drv.ms/b/s!AlzQUCOyZQhKgUPkkH2B2XPhaOSb?e=GZKIlW

Handson Technology, “L298N Dual H-Bridge Motor Driver,” MDU-1049 User Guide, https://www.handsontec.com/dataspecs/L298N%20Motor%20Driver.pdf

HUA HAI JI, “LM2596 Adjustable DC-DC DC to DC Voltage Regulator,” Amazon, https://a.co/d/7tX8dkj

Wemos, “D1 mini Lite,” ESP-8288 D1 Mini Lite User Manual, https://www.wemos.cc/en/latest/d1/d1_mini_lite.html#d1-mini-lite