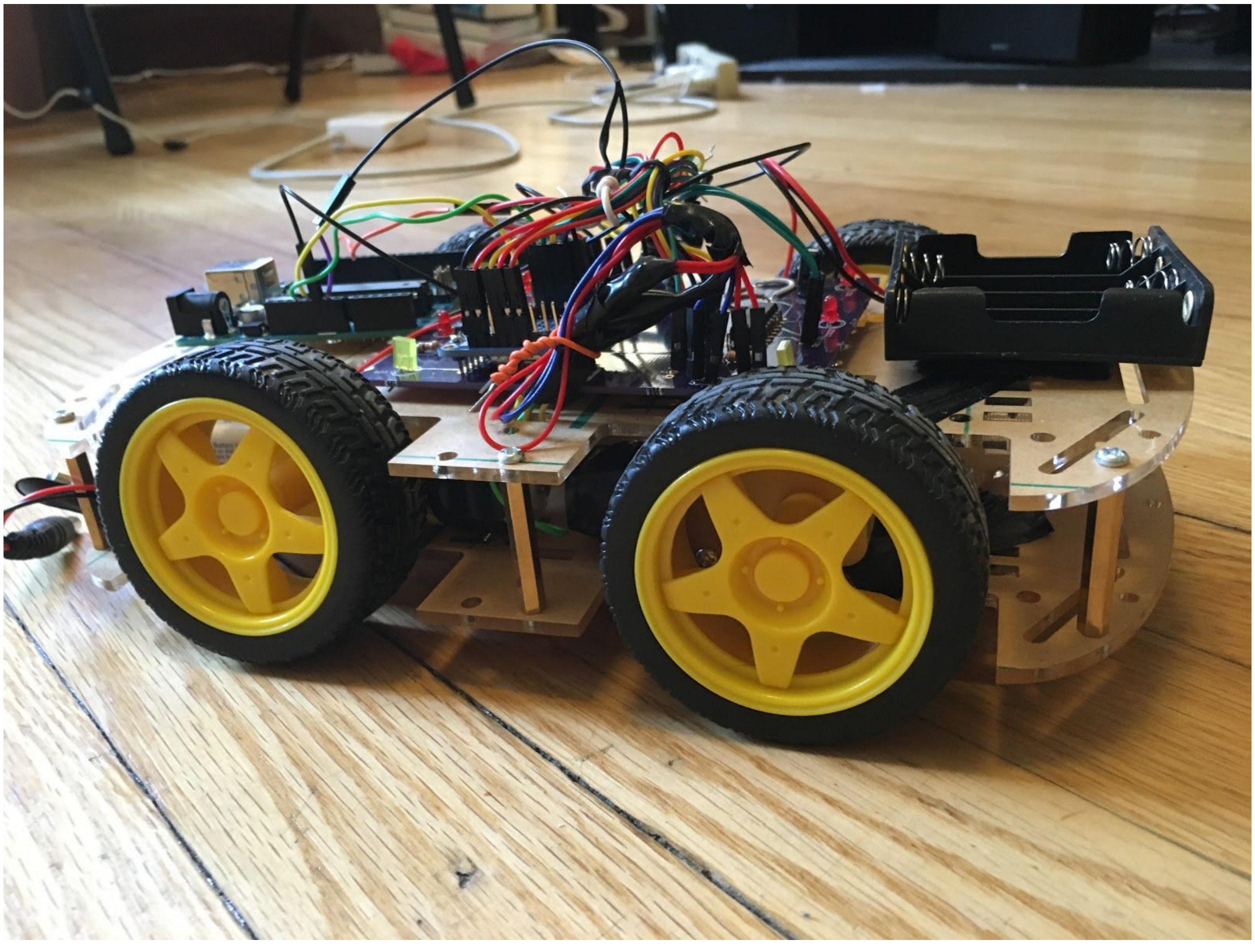
An iPhone Controlled Car

Final Report



The Three Commas:
Bret Pontillo, Alec Taren, & Vansh Patel

Executive Summary:

Our group, The Three Commas, set out to build a toy car controllable with an iphone application. The reason our group wanted to take on this project is that none of us had any experience with iPhone applications. This would allow us to develop skills that could be useful later on in our careers or allow us to pick up a new hobby. Inspiration originally came from the Freescale Cup competition, but we decided to customize the goals.

Our proposed solution involved using multiple different components and bringing them to work together. These include a microcontroller, H-Bridges, motors, printed circuit board, and the phone itself. Originally the microcontroller that would be used predominantly was the Arduino Uno, however the design changed and it was relegated to only powering the h-bridges and two of the LED's.

The car itself was made of a pre-built chassis that we assembled. In order to control the car an iPhone application was used called "Blynk" this would allow the user to send commands via an interface to the car. Receiving the commands was the microcontroller ESP8266 Huzzah. The microcontroller's outputs controls the logic pins on two h-bridges on the car. The signals travel via traces on the PCB, in which most of the other components are located.

The user is able to to move the car in any direction, using four buttons on the iphone application. In addition there are four controllable leds on the PCB of the car. The user can also turn on or off, the h bridge enables on board the car to stop it from moving. In the end our design was successful and the car is both responsive and relatively easy to use.

Problem Statement:

Together The Three Commas, designed a remote control car except it shall be controlled through an iPhone. To control it with an iPhone we configured an application from the Apple App Store - Blynk. The entire car concept is based on the Freescale Cup Challenge. The challenge was building autonomous toy car to follow around a track. The team wanted to wirelessly control a toy car and everyone is an Apple fan, so iPhone control seemed like a perfect match. This was the main modification. It was the utilization of the iPhone to control versus the autonomous controls for the Freescale Cup competition. The Blynk application is a microcontroller control interface that can be designed, configured, and utilized across Apple and Android platforms.

A printed circuit board (PCB) consists of light emitting diodes (LED's), capacitors, resistors, H-Bridge chips, and connections for an Arduino Uno and ESP8266 Huzzah. The printed circuit board incorporates all the different modules and modifications into the design. The car can utilize the LEDs for signaling different steps of the operation such as turns and brakes. The LEDs can be controlled and turned on or off depending on the direction the user would like to signal. There are also two LEDs always on to help determine if the H-bridge chips are being properly powered.

We mainly utilized an ESP8266 Huzzah microcontroller to implement the solutions and design. This ESP8266 is capable of incorporating wireless connectivity and works well with the Blynk application. In addition, it has 9 general purpose input/output pins for the team to write and read to. There is also a voltage regulator pin in and out, to help limit the amount of voltage to power the chip itself.

The software to control the ESP8266 Huzzah microcontroller hardware was written through the free and open source Arduino Software integrated development environment (IDE). Blynk served essentially as the integrated development environment for the team to configure the iPhone application software. Blynk uses more of a visual software configuration, where one can drag and drop widgets. After the widget is added, the output pin can be selected and a function could be added. The development application utilizes a cloud-based server, hosted by Blynk and TCP/IP protocol to send the commands selected on the application to the ESP8266 Huzzah.

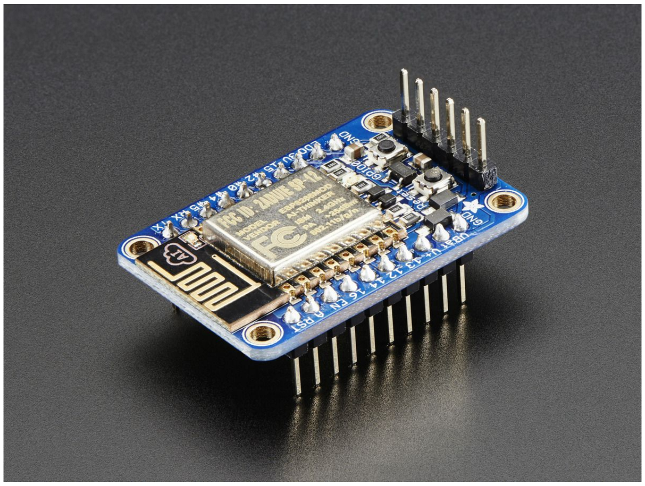
The team planned to use an Arduino Uno as the primary microcontroller. Instead, it was utilized as a voltage regulator due to timing and supplies. This was due to some issues with our design implementation and circuit board. The Arduino Uno powers the H-bridge chips which control the four motors of the toy car. The Blynk application sends messages and the ESP8266 receives the information over the wireless network.

The image 1 and image 2 located in the appendix portion of the report, are conceptual design of what the team was inspired by to do this project. Image 1 shows the Freescale Cup Competition car concept which inspired us. The car has 4 motors, a chassis, circuitry, and tires all exposed. The design we aimed for was to connect all those things through the printed circuit board. Image 2 shows a car being controlled by an application on an iPhone. We wanted to design a similar app to the one shown and be able to control the movement of the car in any direction and control the LEDs representing the turn signals.

Proposed Solution:

Our design plan was a remote control car except it shall be controlled through an iPhone. To control it with an iPhone we use an ios specific app. The entire car concept is based on the Freescale Cup Challenge but with few changes. The main modification is the utilization of the iPhone to control versus the autonomous controls in that competition.

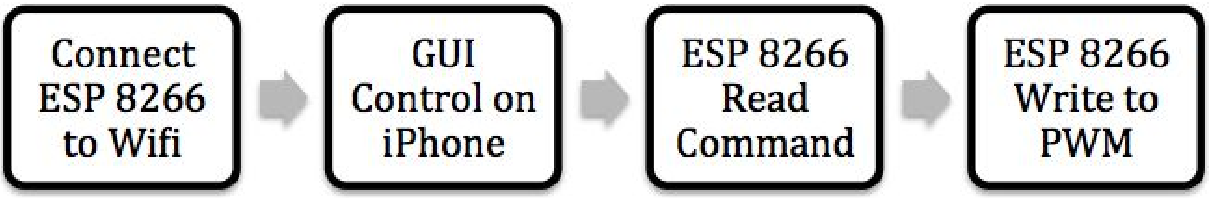
Due to time limitations, instead of creating an iPhone app we used a customizable configuration app to control the car. Our group planned to utilize a wireless module and a SD card module, but due to deadlines and budgetary issues we decided to implement SD card module in future with more time. We also wanted to use arduino to play music and honk the horn but again due to time deadlines we focused our project on the bigger picture for the time being.



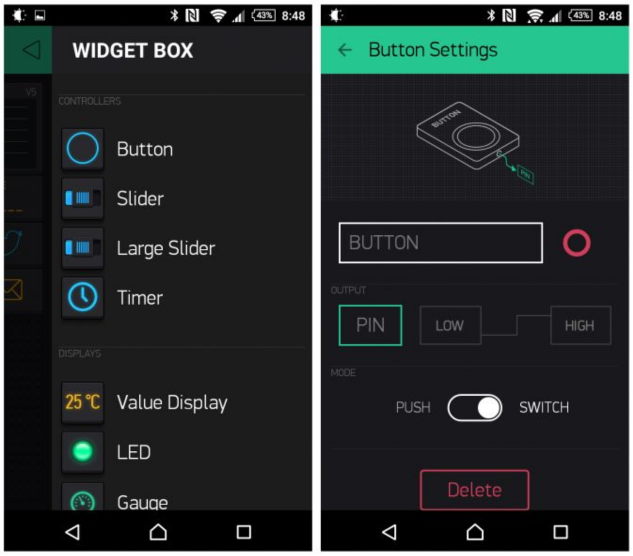
The app we used was blynk, it allowed us to change the PWMs on the huzzah. This app saved us a large amount of time opposed to creating our own application. It allowed us to create an interface that was easy to use. The car had individual wheel motors for all four wheels. The car had six LED’s, four were controllable by the iPhone and two were always on to show the power status of car. Two nine volt batteries and four AA batteries. It also consisted of ESP8266 as a microcontroller and wireless module. The combined weight of the car was roughly four pounds. The app was controlling the movement of car in any direction.. User could also remotely turn on or off the motor logics. The proposed solution was originally much different than what was finally done to complete the project. Originally the microprocessor that we planned on using was the Arduino Uno and Uno is where we would connect all the logic pins of the h-bridge and wireless module. The proposed solution was altered on the ninth week and the team agreed to use ESP8266 as the primary microcontroller to control the h-bridges and the LED's. We decided to use Arduino Uno as voltage regulator for now, but also to allow us to control different sensors in the future.

Implementation Details:

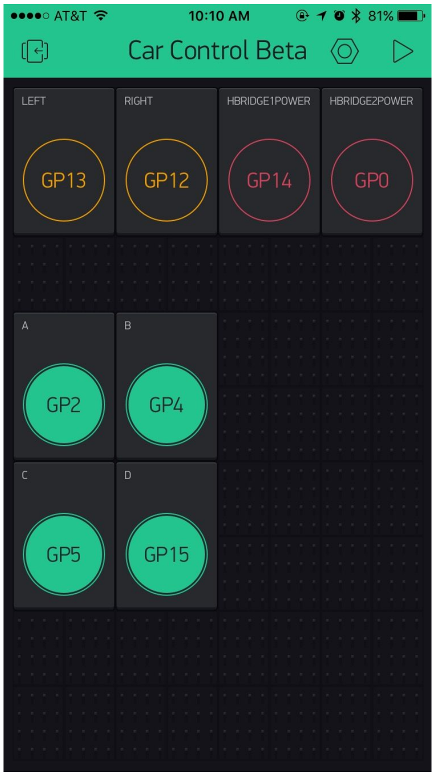
The main microcontroller component used for the design implementation was the ESP8266 Huzzah. The Huzzah is an 80 MHz microcontroller with full front-end Wifi with TCP/IP support. The board on the huzzah consists of two parallel breakouts. It has nine general purpose input/output pins as well as a voltage regulator. The regulator on the chip is important to help regulate the voltage to the 3.3 volt logic level. This is the digital logic level for our PWM outputs on the board.



The ESP8266 Huzzah comes pre-loaded with NodeMCU's Lua Interpreter. The team did not use this for our software. We flashed the microcontroller to be able to use the Arduino integrated development environment to program the chip. To do so, we downloaded the ESP8266 board package for the preferences and to support the chip. A FTDI cable was used to program the Huzzah, using a Baud rate of 115200 and a CPU frequency of 80 MHz. The program uploaded to the Huzzah includes an ESP8266 specific library as well as a Blynk library to use the application. The authentication id is defined to connect with the specific iPhone in use as well as the data rate of 9600 Mbps and the wireless network we are connecting with. The last piece of code runs a loop to stay connected with the Blynk app.

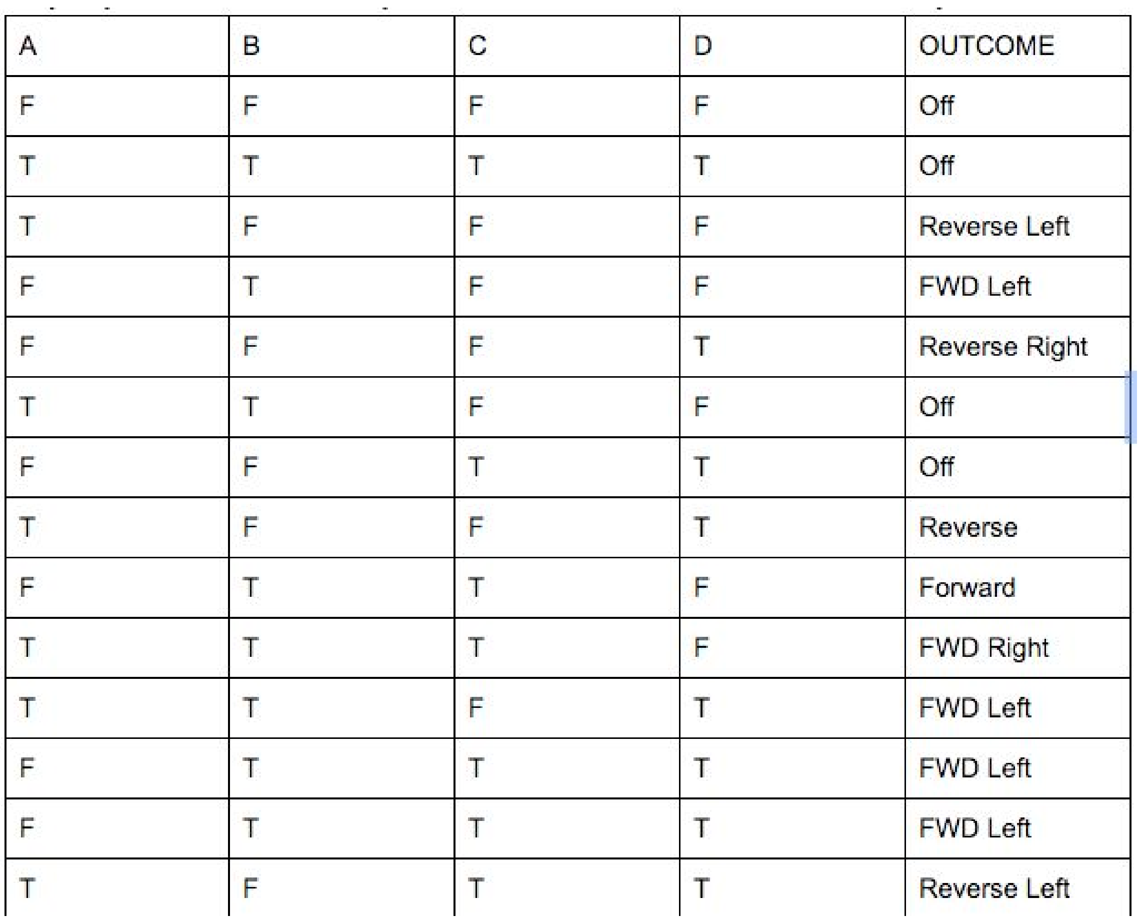


As seen in the picture above, the Blynk application has to be configured to work with the program encoded on the ESP8266. To edit the application, one must click to add a widget. From this, you can select a PWM pin to write to and what the function should be. For all our connections, we utilized button widgets as switches.



Our application in the image above, on the iPhone had 8 buttons to control the car. There are two buttons to control the lights on the PCB board. Pressing the "Left" button writes to GP13 on the Huzzah, this turns on the 2 LEDs on the left side of the PCB board. Pressing the button labeled "Right" writes to GP12, which turns on the 2 LEDs on the right side of the PCB board.

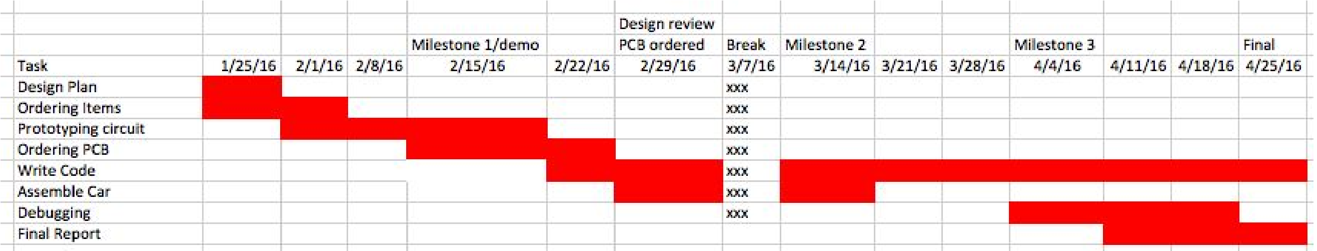
The button "HBRIDGE1POWER," controls the enable of the H-bridge chips. When on, it will write to GP14 and thus allow us to control the logic of the chips, when off we cannot move the car. This along with "HBRIDGE2POWER," which writes to GP0 which controls the second H-bridge, act as a kill switch to power the motion of the car. With these buttons, they enable the use of the motors. The logic in the table below are the different scenarios on the motion of the car when different buttons are on (T = true) or off (F = false). The outcome is based on the front of the car being the location of the Arduino uno.



A	B	C	D	OUTCOME
F	F	F	F	Off
T	T	T	T	Off
T	F	F	F	Reverse Left
F	T	F	F	FWD Left
F	F	F	T	Reverse Right
T	T	F	F	Off
F	F	T	T	Off
T	F	F	T	Reverse
F	T	T	F	Forward
T	T	T	F	FWD Right
T	T	F	T	FWD Left
F	T	T	T	FWD Left
F	T	T	T	FWD Left
T	F	T	T	Reverse Left

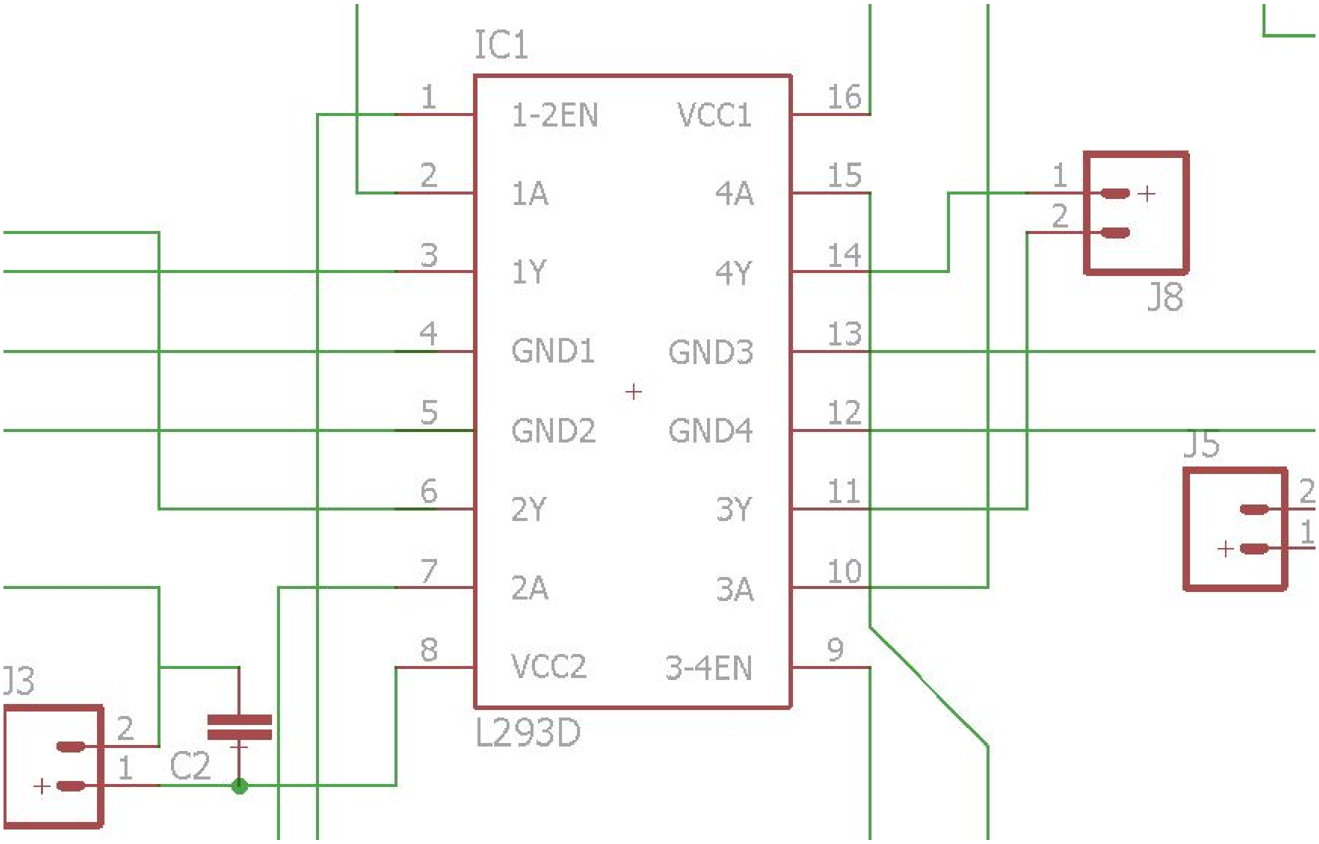
The buttons “A,” “B,” “C,” and “D,” control the logic of the H-bridge chips on the PCB. When the chips are enabled high, the above truth table can then be used to control the motion in any direction of the 2D plane. The buttons “A” and “B” control the logic for H-Bridge 1 and the buttons “C” and “D” control the logic for H-Bridge 2. From testing the controls, the best and easiest way to maneuver the car is to always keep buttons “A” and “D” off and keep “B” on most of the time and just switch “C” on and off determining the direction. Each H-Bridge controls the 2 wheels on each side of the car. To turn off one h-bridge using logic, “A” and “B” would have to be at the same logic level, same goes for “C” and “D.”

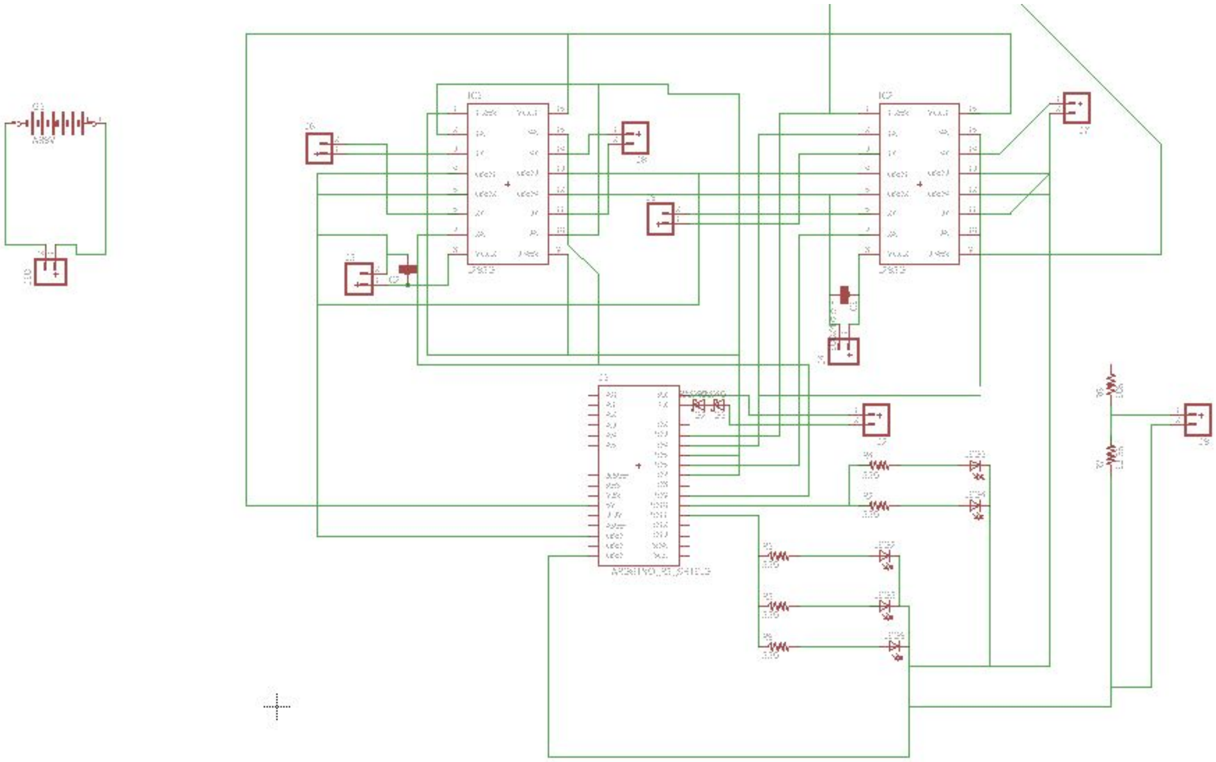
The PCB design involves multiple components, however because of the changes in our design plan some of the PCB connections were changed. In the center of the PCB is the connections for the Arduino Uno, those connections were then soldered to the ESP8266 Huzzah as that was the microcontroller we were going to use. The Arduino is still connected to the PCB ground and both the 5 and 3.3 volt traces. These are powering the the H-bridge chips located on PCB and two red LEDs respectively. The Huzzah's PWMs are connected to the H-bridge enables as well as its logic gates through traces. The Huzzah also has two PWMs that are connected to resistors and LEDS which allow them to be controllable by the user. Connected to the various motors power supplies are capacitors at 10 microfarad. The resistor values on the PCB were at 40 ohms throughout. The motors were connected to the h-bridge through traces. Because of design changes the RX and TX pins were no longer used. Below you can see the overview of the PCB board see appendix.



						Design review								
				Milestone 1/demo		PCB ordered	Break	Milestone 2			Milestone 3			Final
Task	1/25/16	2/1/16	2/8/16	2/15/16	2/22/16	2/29/16	3/7/16	3/14/16	3/21/16	3/28/16	4/4/16	4/11/16	4/18/16	4/25/16
Design Plan	🟥						xxx							
Ordering Items	🟥	🟥					xxx							
Prototyping circuit		🟥	🟥	🟥			xxx							
Ordering PCB				🟥	🟥		xxx							
Write Code					🟥	🟥	xxx	🟥	🟥	🟥	🟥	🟥	🟥	🟥
Assemble Car						🟥	xxx	🟥						
Debugging							xxx				🟥	🟥	🟥	
Final Report												🟥	🟥	🟥

The circuit design has the huzzahs PWM outputs connected to logic pins on the two H-bridges. The four motors are connected to the Y pins as shown in the figure below. Powering the motors is the 6 volt battery packs that connect to VCC 2 on each H-bridge. The 9-volt batteries power the arduino and the huzzah. The PWMs also control the LEDs. The arduino is powering the the two H-bridge chips.


IC1
1
1-2EN
VCC1
16
2
1A
4A
15
3
1Y
4Y
14
4
GND1
GND3
13
5
GND2
GND4
12
6
2Y
3Y
11
7
2A
3A
10
8
VCC2
3-4EN
9
L293D
1
2
J8
J5
2
1
J3
2
1
C2



Conclusion/Results:

In conclusion we were able to meet all our specifications; we were able to control all 4 motors, we had six LED's with battery life of over 2 hours. We also used an iPhone app to control the movements of the car. The app and the Huzzah responded very quickly to commands over internet. We also used the iPhone app to turn lights and motors on or off.

If we had more time we would definitely fix our PCB and reduce the size of it. We would get rid of Arduino and put better LED's with higher budget. We would also like to implement a joystick on the app to make user controls even more user friendly.

In future we would like to also implement a raspberry Pi so with its own server we don't have to tether on an iPhone when we want to use the car. We would also like to integrate more sensors like a camera for live streaming, a temperature sensor, a speaker for horn or music. Our overall project experience was pretty academically challenging, because we were working with different people on project that taught us a lot more about microcontrollers and different things we can do with it.

Appendices:

Image 1: Car Design Concept

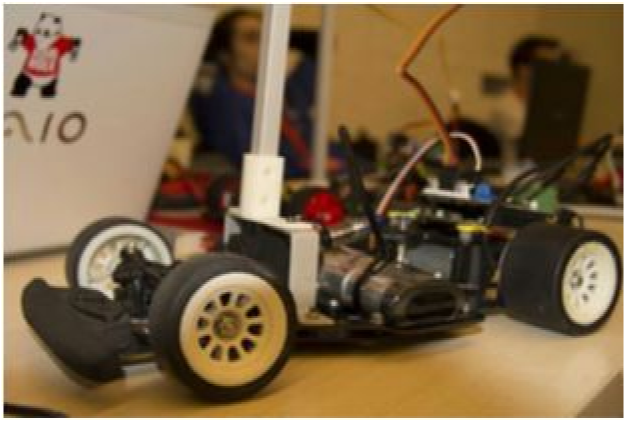


Image 2: iPhone Control Concept