Laser Tag Droid

Jake Hamill, Martin Litwiller, Christian Topete
ECE 445 Design Document

Teaching Assistant: Eric Clark
February 24, 2017

Table of Contents

1. Introduction

1.1 Objective

Our proposed project is to design, build, and test a remote control laser tag droid that allows users to engage in exciting physical and computer gameplay. When switched into autonomous mode the droid will be able to search for, identify, and engage opponent droids automatically. This will create a fun laser tag experience whether you are playing solo or with a group. It will be a wheeled vehicle with the mobility to move around quickly and maneuver easily. The droids will be controllable wirelessly via an iOS app which allows users to play the game. (Note that we are only building one droid for this semester project.)

1.2 Background

Over the past 5 to 10 years, the market for remote control toys of all types has expanded dramatically. With the technology steadily becoming less expensive and more accessible, these toys have gotten a huge amount of exposure and have gained in popularity with consumers of almost every age. However, most products in this category are simple models of cars, planes, or helicopters that are only meant to be driven or flown around and nothing more. A relatively small number of RC toys take advantage of their potential to be used as (or in) games. Typically if there is a game or competition involving RC toys, it consists only of racing. However, many people are naturally competitive and would prefer something more dynamic than simply driving or flying. One example of a very popular type of RC competition are battlebots. The drawback to these competitions is the inevitable and significant damage that the vehicles suffer. Therefore, we believe that an alternative non-destructive type of RC competition such as laser tag would be very popular with consumers. Laser tag bots have been implemented in several ways already. Some have been bipedal walking robots, and some have been built with a simple wheeled car design. Still others have been implemented using flying quadcopters [3]. Even with these, however, there are drawbacks. Flying vehicles are somewhat hard to control and slightly dangerous, and bipedal or wheeled designs are often rather slow and unexciting. Our design will attempt to remedy these negative factors and add unique functionality that would make for a successful product.

1.3 High level Requirements

- The droid will have an autonomous target seeking mode, in which it will move around on its own looking for targets.
- The droid will have a player-controllable mode, in which its movement and firing can be controlled by a user with a mobile phone app.
- The droid will be powered by a standard 7.4V LiPo hobby battery.

2. Design

2.1 Block Diagram

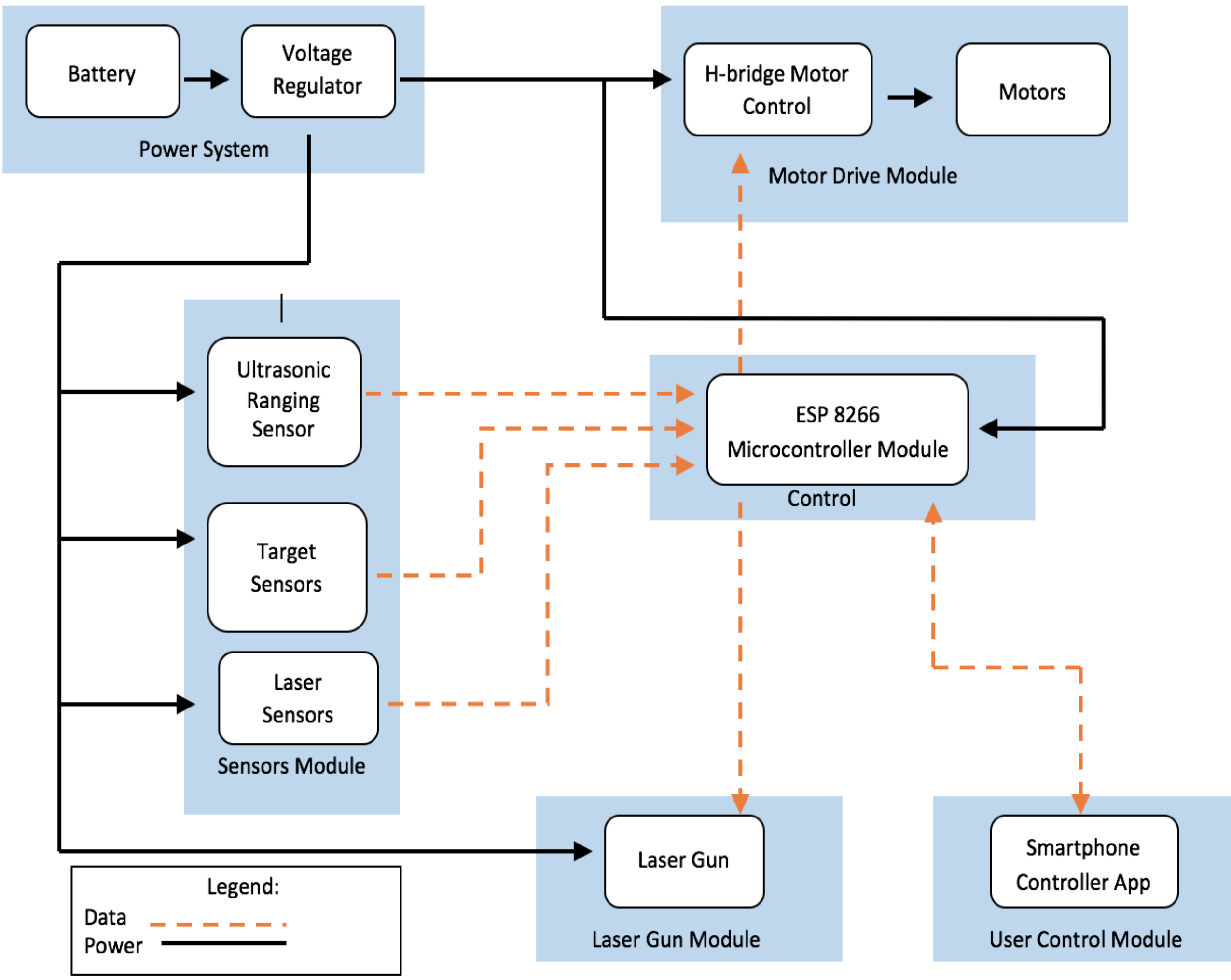

Figure 1: Top Level System Block Diagram

Power System: Battery, Voltage regulator
Motor Drive Module: H-bridge motor control, DC brushed motors
Control Module: ESP 8266 Microcontroller Module
Sensor Unit: Ultrasonic Range Sensors, Laser Sensor, Target Sensor
Laser Gun Module: 5mW Red Laser Pointer
User Control Module: Smartphone Controller App

2.2 Physical Design

The vehicle chassis will be built from layers of sheet metal connected by standard hex stand-offs. Motor gearbox kit, wheels, and sensors will be mounted on lower level, while the microcontroller and PCB will be mounted on upper level. The overall 3-D physical dimensions will be approximately 6x4x4 (WxLxH) inches. An array of photodiodes for detecting targets will be mounted around the circumference of the vehicle on the bottom layer. Several photodiodes for sensing incoming laser fire will be mounted on the top layer. Three sonar ranging sensors will be mounted on the front, left side, and right side on the bottom layer. The laser will be mounted on the top layer facing forward.

2.3 Power Supply

The power supply is essential in keeping all systems working efficiently and safely. The power supply will consist of one KA378R05 low dropout voltage regulator. The voltage regulator will be used to output 5 volts to each DC motor and to the sensor module. These low voltage dropout regulators will work well with the 7.4 volt LiPo battery due to the low voltage drop from 7.4 to 5 volts. Additionally, the voltage regulator max current rating is 3 amps but we will limit it to 2.5 amps so we can prevent the regulator from overheating. We will be adding a heat sink to the regulator for it to be able to dissipate the heat much quicker. The heat sink allows the regulator to dissipate a maximum of 15 watts.

2.4 Battery

The battery is a standard 25C 7.4 Volt 2500 mAh battery [1]. This is a 2 cell battery which outputs a nominal 3.7 volts per cell. Its peak voltage is 8.4 volts and minimum is 6.4 volts. The max current is 62.5 amps which bodes well with our circuitry. This will power the microcontroller, two DC brushed motors, and the sensor modules. We will also be implementing a battery protection circuit to ensure that the battery will not get damaged from over discharging. The protection circuit involves 5 resistors, 1 TL431 adjustable shunt regulator, and a SFT 1350 P-Channel Power MOSFET. R1 is used to pull up the output when the TL431 is off and to turn off the SFT1350 p-channel MOSFET causing the circuit to cutoff. R2 and R3 is used to adjust the voltage at which we want the circuit to cutoff. R4 is used to eliminate hysteresis and R5 is used as a load. The microcontroller will poll the voltage output and will shut down the droid if the battery overheats.

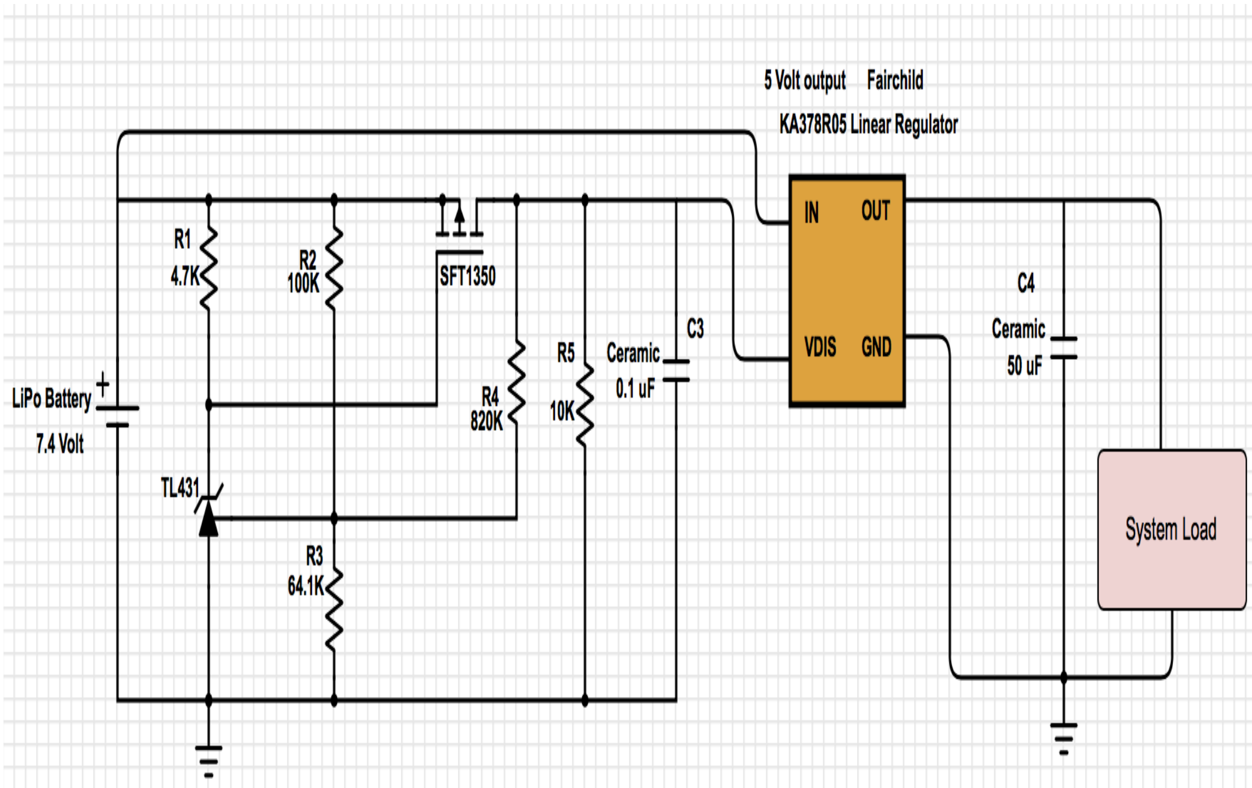

Figure 2: Power Block Schematic

2.5 Motors

The two motors are going to be DC brushed motors. Each motor has a stall current of 800 mA. They will be powered by 5 volts. We will also be implementing PWM and will be creating an H-bridge circuit in configuration with these motors. We will be using a Tamiya 70097 motor gearbox kit to fit these specifications.

2.6 H-Bridge Motor Control/PWM

We will be implementing the H-bridge circuit as well as the PWM signal from the microcontroller through the SN754410 quadruple half-h driver chip. The H-bridge circuit allows us to turn in a smooth manner while the PWM control will allow us to control the speed of the droid. We will be using two SN754410 chips, one for each motor. Each chip has a max current rating of 1 amp which bodes well with the DC brushed motors that have a stall current of 800 mA. The pwm signal will be through the analog/pwm pins on the microcontroller. We will be implementing 3 modes of speed (slow, medium, fast) for the PWM. The motor will require 2 PWM signal pins and 1 digital. The hardware and software implementation of the H-bridge is shown below with each function. We note that this chip can withstand two motors but due to the stall current rating of the motors, we have to use one chip for each motor.

Pin	Name	H-bridge Function	Functionality
1	1,2 EN	Enable Motor 1	Setting pin to low (0 volts) turns motor off while setting it to high (5 Volts) will turn on the motor.
2	1A	Motor 1 Forward	Connected to PWM pin to change variability of speed in the forward direction.
3	1Y	Motor 1 power	Connect to one of the pins of motor 1
4	GND	Ground	Connect to Ground
5	GND	Ground	Connect to Ground
6	2Y	Motor 1 power	Connect to the other pin of motor 1.
7	2A	Motor 1 reverse	Connect to pwm pin to change variability of speed in the reverse direction.
8	VCC2	Motor 1 power source	Positive power source for motors. It will be 5 volts in this case.
9	3,4 EN	Enable Motor 2	Setting pin to low (0 volts) turns motor off while setting it to high (5 Volts) will turn on the motor.
10	3A	Motor 2 Forward	Connected to PWM pin to change variability of speed in the forward direction.
11	3Y	Motor 2 power	Connect to one of the pins of motor 2
12	GND	Ground	Connect to ground.
13	GND	Ground	Connect to ground
14	4Y	Motor 2 power	Connect to the other pin of motor 2
15	4A	Motor 2 reverse	Connect to pwm pin to change variability of speed in the reverse direction.

Pin	Name	H-bridge Function	Functionality
16	VCC1	Motor 2 power source	Regulated 5 volts

Table 1: H Bridge Chip Pinout

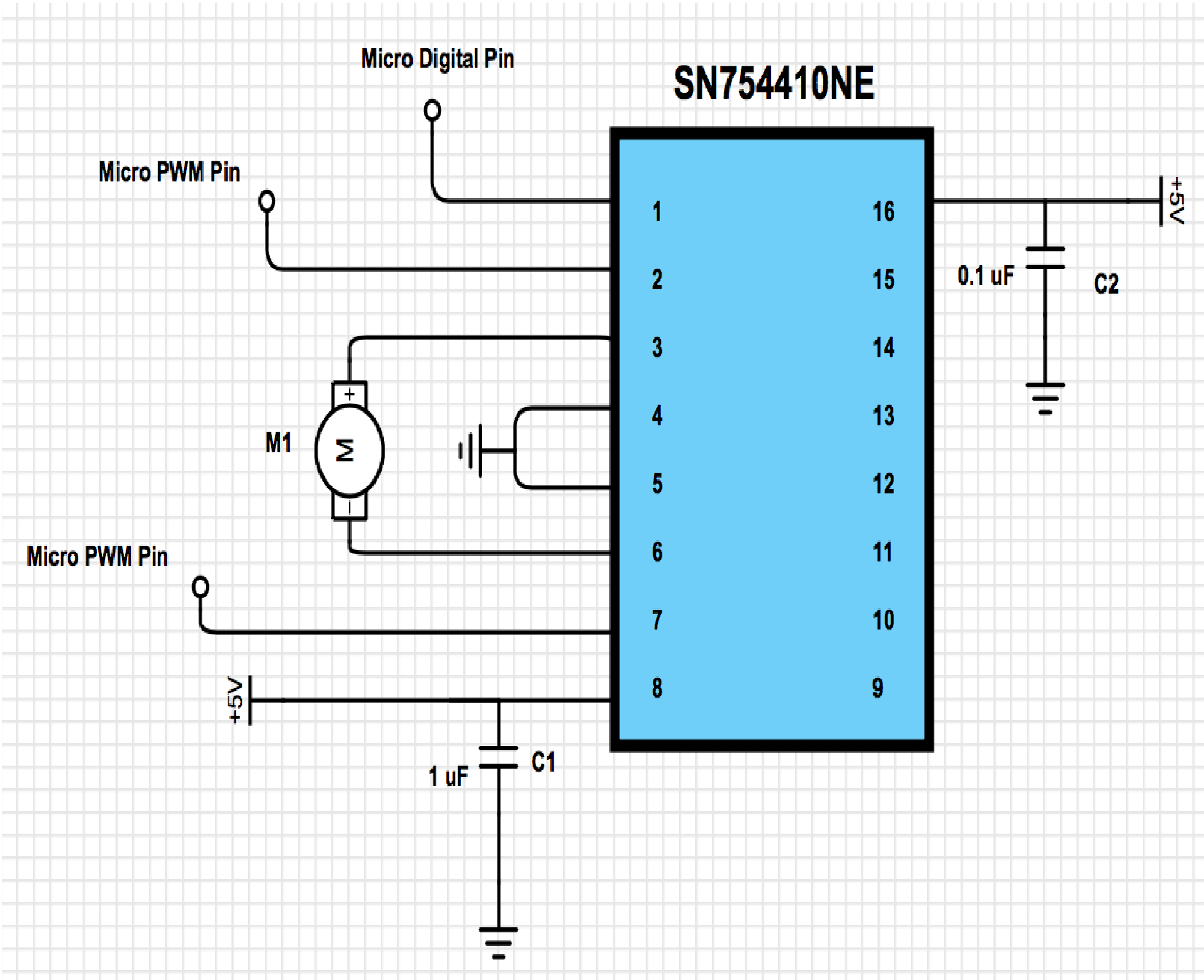

Figure 3: H-bridge Circuit for one motor

2.7 Ultrasonic Range Sensor Module

We will be using the HC SR-04 ultrasonic range sensor on the outside of our droid to help with the autonomy of the droid. The ultrasonic range sensor sends out a signal and then waits for the signal to determine how long it took. The sensor will then convert this time into a distance which helps us determine the proximity of any nearby objects. The sensor can determine obstacles within a range of 2cm to 400cm. This module will be powered by 5 volts and controlled by the microcontroller.

2.8 ESP8266 microcontroller/wifi module

This device is a combined microcontroller and wifi communications module. We will use the microcontroller portion to poll all sensors and control all parts of the vehicle such as the motor control circuitry and the laser gun. It will use pins connected to the circuitry to control the motors, laser, etc. The ESP8266 will act as a server, and when a client (mobile app) connects to it, it will take control commands from the app (see section 2.11). The ESP8266 will also be programmed for controlling the droid when it is in autonomous mode (High Level Req. 1). If the user sets the droid to manual mode via the app, then the ESP8266 will listen to the steering commands sent by the app and will set the pins accordingly. Figure 4 shows a high level flow chart of the navigation logic.

The autonomous logic will be developed through experimentation and testing over the course of our project. The goal of the logic is to enable the droid to roam around some area, aiming towards (and firing at) targets if it sees any, and avoiding obstacles (e.g. not running into walls, etc.). It will use the Ultrasonic Range Sensors (2.7) and the Target Sensor Module (2.10) to sense and avoid obstacles as well as seek targets. Figure 5 shows an overly simplistic example of what the autonomous logic could look like (in reality, we will develop a longer and more advanced algorithm).

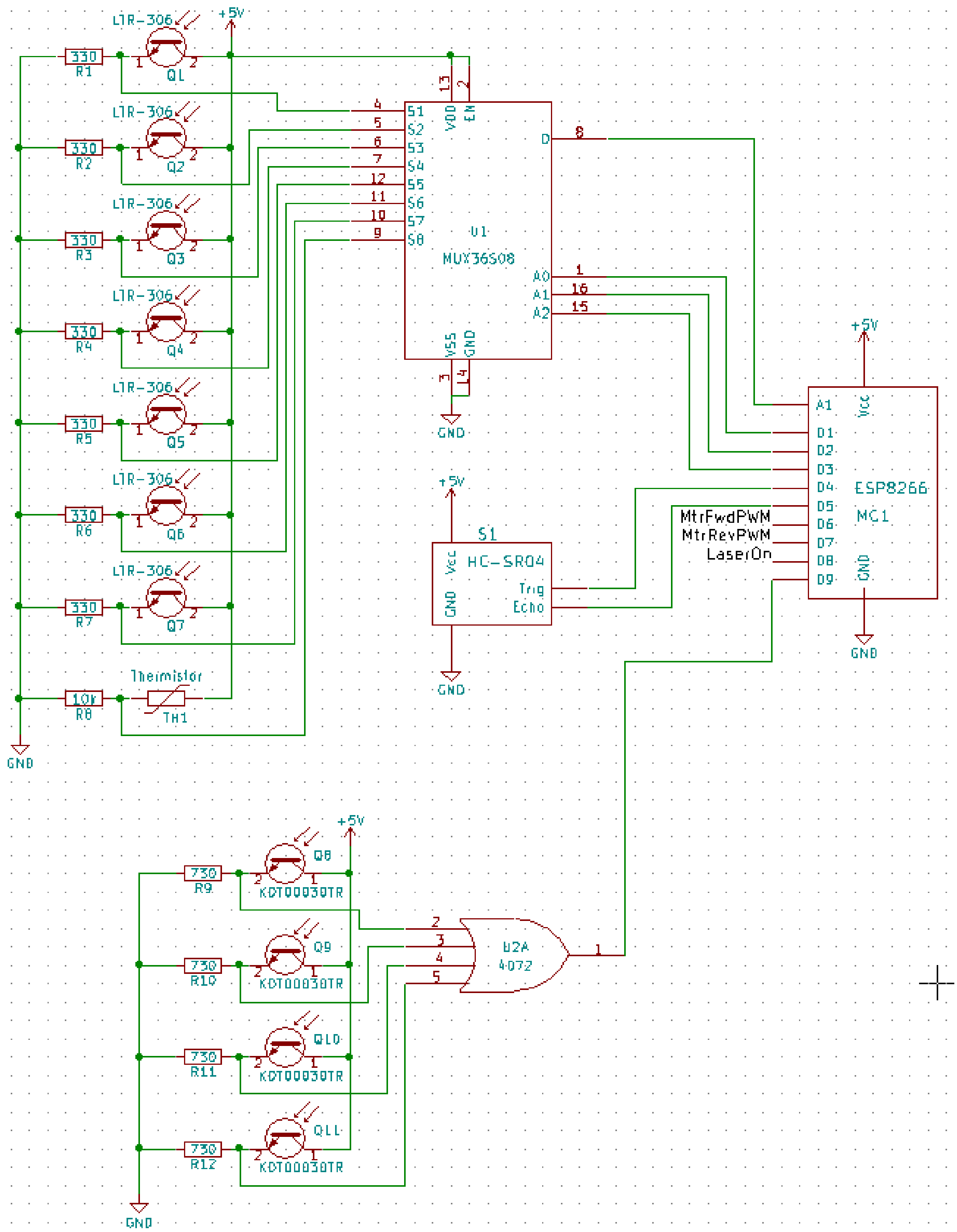

Figure 4: Sensor Module and Control Module Schematic

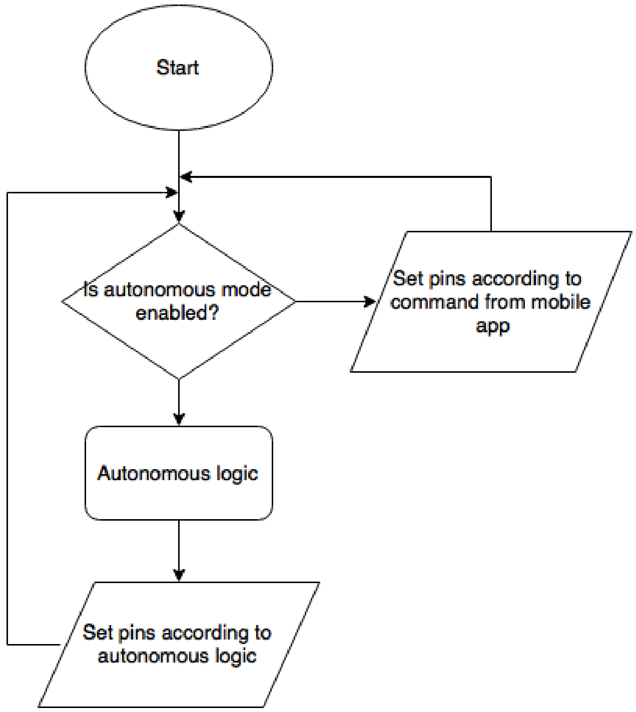

Figure 5: ESP8266 droid steering high level flow chart

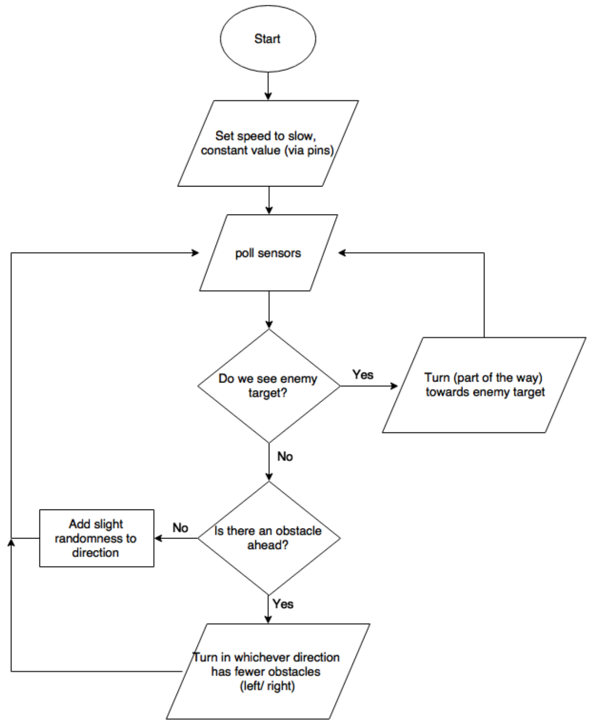

Figure 6: Overly simple example of autonomous logic

2.9 Laser Sensor Module

To detect incoming laser fire from opponent droids, we will use a set of simple phototransistor circuits as shown in figure 4. A phototransistor will generate a current through its emitter resistor if hit with a red laser beam, which will generate a 3.5-5V output signal. The outputs from the 4 phototransistor circuit copies will be passed through a 4-input OR gate whose output will be connected to the microcontroller. The microcontroller will detect the rise in output voltage as a digital signal and will deduce that the droid has been hit!

2.10 Target Sensor Module

The target sensor module is similar to the laser sensor module. It will use an array of 7 IR phototransistors to generate analog voltage readings, which will be passed to the microcontroller through an analog multiplexer chip. The microcontroller will translate the analog voltage level into relative levels of IR light in each direction. If a significant level of IR light is detected, the droid will assume that it is an opponent droid and will turn toward and follow it.

2.11 Mobile Application Module

Functional Overview:

The mobile phone application will enable a human user to control the droid. Through the app, the user will be able to accelerate the droid forwards and backwards, as well as turn to the left and to the right (High Level Req. 2). The user will also be able to fire the laser from the app. Furthermore, the app will allow the user to toggle between autonomous driving mode and manual (driven by the user through the app) mode (High Level Req. 1, 2).

More specifically, the mobile app will communicate with the droid over Wifi. When the user taps in a specific region of the screen (the “shoot” button), the app will send a “shoot” signal to the droid which will cause the laser to be fired. The droid will have two modes of operation: an autonomous mode, and a manual mode. There will be a button in the app (“toggleMode”) that allows the user to toggle between the two different modes. When the toggleMode button is pressed, the app will switch modes.

In manual mode, the user will steer the droid by tilting the phone in the desired direction of motion. More specifically, there are 3 different measures that will define it’s position in : yaw, pitch, and roll (see figure 7).

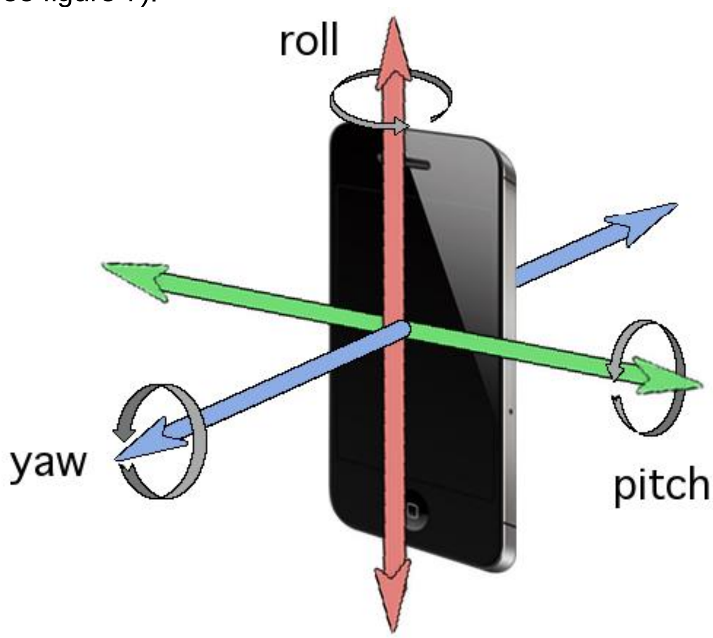

Figure 7: visual description of phone position in 3d space [6]

For purposes of this explanation, define yaw, pitch and roll to all be zero when the user is holding the phone horizontally with the screen perpendicular to the ground and the home button on the user’s right (Screen facing user). We will say that yaw, pitch, and roll are positive if the user rotates the phone slightly (less than 90 degrees) in the direction of the arrows in figure 7 and negative if the phone is rotated slightly in the opposite direction when starting from zero.

We care in particular about the roll and yaw. The roll will define the forward/ backward velocity, and the yaw will define the degree to which the droid is turned to the right or left. There will be a tolerance region (ε) around zero for each value. If the angle is within ±ε of 0, it will be considered zero, thus making it easier for the user to keep the droid stationary. Furthermore, rotations of more than 90˚ (positive or negative) will be considered “undefined” — that is, there will be no speed value associated with these regions. All speeds (in the case of roll) or steering angles (yaw) will be determined by discretizing the regions $(90 - \varepsilon)$ into evenly spaced intervals, where each interval represents one speed. For example, if there are r possible speeds (or angles of turning), then the size of each interval, s, is given by:

$$s = \frac{90 - \varepsilon}{r} \tag{1}$$

and then the interval of the nth smallest value (e.g. speed) begins at $s(n-1)$ degrees and ends at sn degrees. So each of these n intervals will have a specific speed or direction value associated with it. Having a large number of speed/ direction intervals will give the user a more smooth steering feel. The flowchart describing the "steering" loop of the app is shown in figure 8 below. This loop can be interrupted by pushing the toggle mode button (thereby switching the app into autonomous mode).

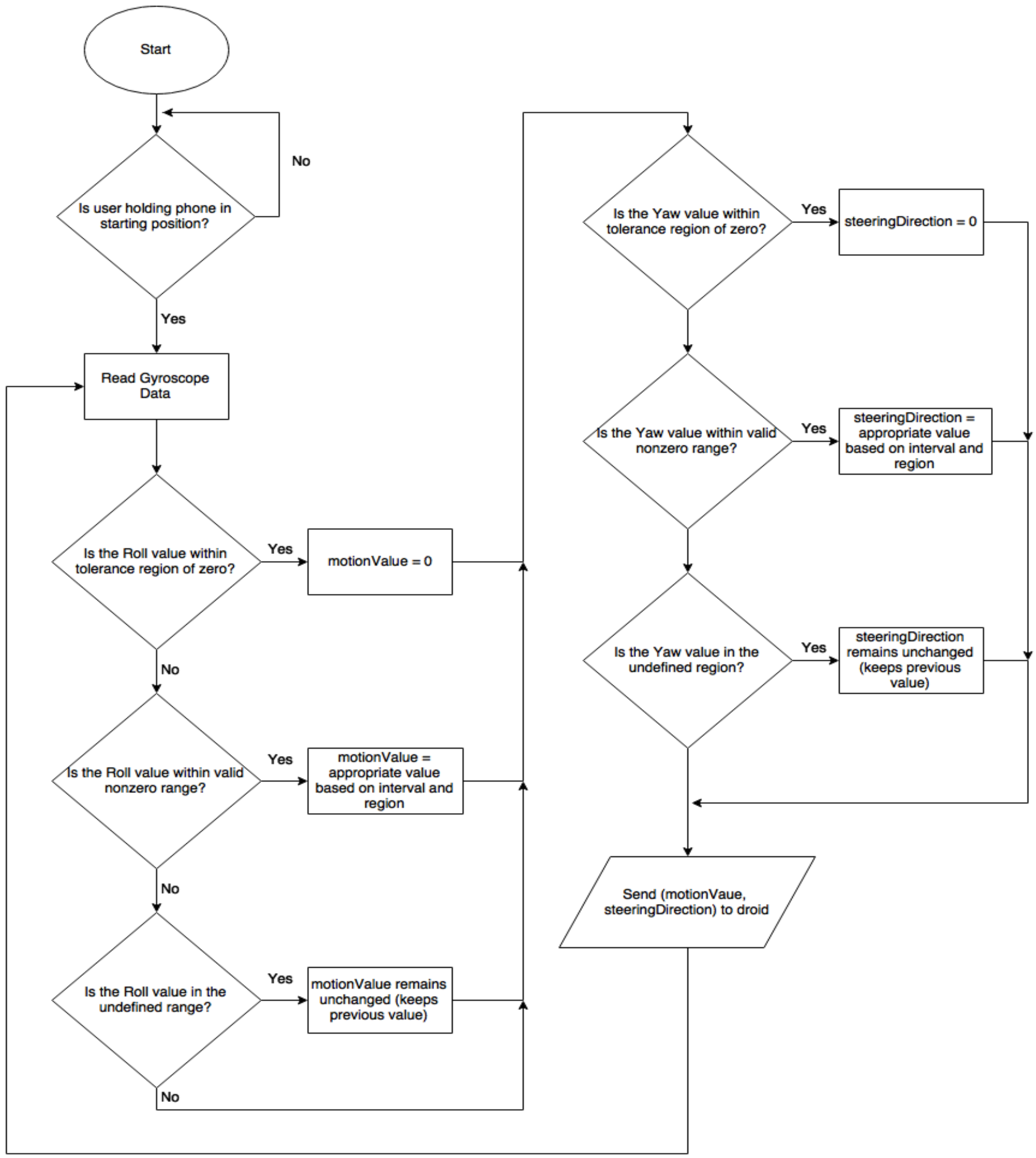

Figure 8 Mobile App steering logic flow

3. Calculations and Simulations

3.1 Cutoff Voltage Calculation

In order to ensure our circuit cuts off the LiPo battery at 6.4 volts, we need to calculate resistance values for both R2 and R3. Our TL431 has a reference voltage of 2.5 volts and we calculate the cutoff voltage based on the equation below.

$$V_{cutoff} = V_{ref}\left(1 + \frac{R_2}{R_3}\right) \qquad (2)$$

We then determine the relationship between R2 and R3 by having a cutoff voltage of 6.4 volts and a reference voltage of 2.5 volts.

$$6.4 = 2.5\left(1 + \frac{R_2}{R_3}\right) \qquad (3)$$

$$1.56 * R_3 = R_2 \qquad (4)$$

We pick large arbitrary values for R2 and R3. We pick values in the kilo-ohms to ensure each resistor isn't dissipating a lot of power. The values we selected are R2=100 Kilo-Ohms and R3=64.1 Kilo-Ohms.

3.2 Minimum and Average Lifetime of Battery

Next, we want to calculate the average lifetime of our battery when our values are at average rating and max rating. Each component and its corresponding current rating are shown in the table below.

Module	Average (mA)	Max (mA)
ESP 8266	250	500
DC motors (per)	400	800
Ultrasonic Sensor (per)	15	15
KA378R05 Voltage Regulator	10	10
Phototransistors	10	21.2
Laser	30	30

Table 2: Module current consumptions

We first calculate the average lifetime of the battery. We calculate the total average current and then divide it by the battery's capacity to determine the average run time.

$$I_{Total} = 2 * I_{per\ motor} + I_{ESP8266} + I_{laser} + 3 * I_{per\ range\ sensor} + I_{phototransistors} + I_{Voltage\ Regulator} \quad (5)$$

$$I_{Total,Avg} = 2(400) + 250 + 30 + 3(15) + 10 + 10 = 1145\ mA \quad (6)$$

$$Capacity = 2500\ mAh \quad (7)$$

$$Average\ Run\ time = \frac{Capacity}{I_{Total,Avg}} = \frac{(2500\ mAh)}{1145\ mA} = 2.18\ hours \approx 131\ minutes \quad (8)$$

We see that the average lifetime of our battery will be approximately 131 minutes. Next we want to determine the minimum lifetime of our battery by using our max ratings.

$$I_{Total} = 2 * I_{per\ motor} + I_{ESP8266} + I_{laser} + 3 * I_{per\ range\ sensor} + I_{photodiodes} + I_{Voltage\ Regulator} \quad (9)$$

$$I_{Total,Max} = 2(800) + 500 + 30 + 3(15) + 21.2 + 10 = 2206\ mA \quad (10)$$

$$Capacity = 2500\ mAh \quad (11)$$

$$Max\ Run\ time = \frac{Capacity}{I_{Total,Avg}} = \frac{(2500\ mAh)}{2206\ mA} = 1.13\ hours \approx 68\ minutes \quad (12)$$

We see that the time we can run the system at maximum ratings is approximately 62 minutes. This bodes well because we want our vehicle to last us at least 15 minutes

3.3 Power dissipation of voltage regulator

Next, we would like to calculate the power dissipation of the KA378R05 voltage regulator to ensure that it can withstand the system's load.

$$V_{IN,Max} = 8.4 \qquad V_{out} = 5$$

$$P_{Dissipation,Max} = (V_{IN,Max} - V_{Out}) * I_{Total,Max} = (8.4 - 5) * 2435 \quad (13)$$

$$P_{Dissipation,Max} = 8.28\ Watts \quad (14)$$

We note that our regulator can handle the system load because it has a max rating of 15 watts with the heat sink.

4. Requirements and Verifications

4.1 Mobile App

Requirement	Verification
2.11 Mobile Application Module 1. Pressing of both the “toggleMode” and “shoot” buttons successfully send their respective control messages over wifi	Equipment: Laptop, internet access Procedure: A) set up test server on laptop B) connect app to test server C) press both buttons D) confirm test server received correct control signals Presentation of Results: 2 booleans indicating successful reception the 2 signals
2.11 Mobile Application Module 2. When in manual mode, the app correctly sends correct control signals motionValue and steeringDirection, which correspond to forward/ backward, left/ right (at varying intensities) based on the rotation of the device in the respective directions	Equipment: Laptop, internet access Procedure: A) set up test server on laptop B) connect app to test server C) tilt phone forward, then backward, then rotate right (like steering wheel) and left. D) confirm test server received correct control signals upon each movement Presentation of Results: 4 booleans indicating successful reception of forward, backward, left, right signals
2.11 Mobile Application Module 3. Latency - droid must be able to initiate change in velocity within .25 seconds of phone rotation	Equipment: phone, droid, stopwatch or slo-mo camera Procedure: A) position stationary droid at user’s feet, with user holding phone in hands ready to steer B) user tilts phone forward, initiating the forward motion signal to be sent to the phone, and timer starts timing C) timer stops timing when car starts moving forward Note: this may be below human reaction time, so can also use slo-mo camera to measure time. Presentation of Results: A single real number indicating the time between phone rotation and droid movement

4.2 Microcontroller

2.8 ESP8266 microcontroller/ wifi module 1. Target acquisition - droid must be able to sense target, turn towards it, and shoot.	Equipment: droid, target Procedure: A) place the droid on the ground, and place the target on the ground 10 feet behind the droid B) enable autonomous mode on the droid C) confirm that the droid turns around and fires at the target Presentation of Results: 2 boolean values, indicating that the droid successfully turned and successfully fired.

4.3 Power Supply

Requirement	Verification
1. Voltage regulator must be able to output 4.9 to 5.1 volts with a maximum current rating of 3 amps.	Procedure: A) Attach 1.66 ohm power resistor as load B) Attach oscilloscope across load C) Verify that voltage is steady between 4.9 to 5.1 volts
2. LiPo battery must output 7.3 to 7.5 volts with a maximum current rating of 3 amps.	Procedure: A) Attach 2.47 ohm resistor as load B) Attach oscilloscope across load C) Verify that voltage is steady between 7.3 to 7.5 volts
3. Ensure that the battery protection circuit cuts off when the battery reaches below 6.4 volts.	Procedure: A) Detach voltage regulator from circuit and LiPo battery from circuit B) Apply 6.5 volts from the DC power supply where the battery was connected C) Measure voltage across R5 and ensure the voltage is between 6.4-6.6. D) Apply 6.3 volts to from the DC power supply where the battery was connected E) Measure voltage across R5 and verify that the voltage is 0
4. The overall system battery life should at least be 15 minutes	Procedure: A) Upload system with proper software and let the vehicle run in autonomous mode B) Let the droid run for at least 15 minutes C) Verify that the droid is still running after 15 minutes D) Repeat this procedure 3 times to ensure consistency of verification

4.4 Motor Control

Requirement	Verification
1. H-bridge must be able to control the direction of each motor	Procedure: A) From the microcontroller, output high (5 volts) to pin 2 on chip and low (0 volts) to pin 7 B) Verify motor is spinning in the forward direction C) From the microcontroller, output low on pin2 and high to pin 7 on chip D) Verify motor is spinning in the reverse direction
2.H-bridge must control variability of speed	Procedure: A) From the microcontroller, output low (0 volts) to pin7 B) Output 30, 60, 90% duty cycle PWM signal in sequential manner to pin 2 C) Verify that the motor gets faster with the sequential increase in the duty cycle
3. H-bridge must be able to supply 4.9 to 5.1 volts at 800 mA for each motor	Procedure: A) Attach 6.5 ohm resistor as load B) Attach oscilloscope across load C) Verify voltage is steady between 4.9 to 5.1 volts D) Repeat this procedure for other motor

4.5 Sensors Module

Requirement	Verification
1. Laser sensors must output <0.8V under normal operation and >2V when hit with a laser.	A) Power circuit B) Use voltmeter to measure output with no laser C) Use voltmeter to measure output while exposing the sensor to the laser beam.

2. Target sensors must output a range of voltages between 0-1V corresponding to relative levels of light.	A) Power circuit B) Use voltmeter to measure output under only ambient light. C) Use voltmeter to measure output while exposing phototransistors to IR light source in close proximity.
3. Microcontroller should be able to detect the wall within 10 cm of it with an error of 1 cm based on the HC-SR04 readings.	A) Place vehicle 12 cm away from a wall B) Ensure that the vehicle doesn't make an attempt to turn away from wall C) Place Vehicle 10 cm away from the wall D) Ensure that the vehicle turns away from wall We keep in mind that H-motor control implementation will be done before Ranging Sensor Implementation

4.6 Laser Module

Requirement	Verification
1. Laser must turn on when powered with 5V DC.	A) Power circuit B) Visually check if laser is on.

5. Tolerance Analysis

Battery Protection Circuit

In order for the system to function safely, we need to ensure that the battery's voltage doesn't drop below 6 volts. We need to take into consideration the tolerances for the resistances in the battery protection circuit to ensure that the battery never drops below 6 volts even though our calculated cutoff is at 6.4 volts. We will vary the tolerance of the resistive values that give us 6.4 volts of cutoff. By doing this, we are ensuring that even though we don't have the exact resistance, a positive or negative change in our resistor values still allow our battery protection circuit to function safely.

There are multiple resistor tolerances but we will test it with a 1%, 5%, and 10% tolerance. We will be manipulating the values of R2 and R3 to ensure that the battery voltage cutoff does not fall below 6 volts. We note that the original voltage cutoff we will implement is 6.4 volts. We note that our R2 value is 100 kilo-ohms and the R3 value is 64.1 kilo-ohms with the cutoff voltage set at 6.4 volts.

Resistive Tolerance Values						
Original Resistor Value	+1% Tolerance	+5% Tolerance	+10% Tolerance	-1% Tolerance	-5% Tolerance	-10% Tolerance
R2 = 100 K	101 K	105 K	110 K	99 K	95 K	90 K
R3 = 64.1 K	64.74 K	67.3 K	70.51 K	63.46 K	60.9 K	70.51 K

Table 3: Resistive Tolerance Values

We will apply the opposite tolerance for each resistance tolerance value. For example, if R2 has +1% tolerance then R3 has -1% tolerance because if we use both the same tolerance for each resistor, then the tolerances will cancel in our calculation. Our calculations will consist of R2+1% and R3-1% then R2-1% and R3+1%. We do this combination for the 5% and 10% tolerance as well.

$$V_{cutoff}=V_{ref}\left(1+\frac{R_2}{R_3}\right) \quad (15)$$

<u>R2-1%, R3+1%:</u>

$$2.5\left(1+\frac{99}{64.74}\right)=6.32>6 \quad (16)$$

<u>R2+1%, R3-1%:</u>

$$2.5\left(1+\frac{101}{63.46}\right)=6.48>6 \quad (17)$$

<u>R2+5%, R3-5%:</u>

$$2.5\left(1+\frac{105}{60.9}\right)=6.81>6 \quad (18)$$

<u>R2-5%, R3+5%:</u>

$$2.5\left(1+\frac{95}{67.3}\right)=6.03>6 \quad (19)$$

<u>R2+10%, R3-10%:</u>

$$2.5\left(1+\frac{110}{57.7}\right)=7.26>6 \quad (20)$$

<u>R2-10%, R3+10%:</u>

$$2.5\left(1+\frac{90}{70.51}\right)=5.69<\ 6 \quad (21)$$

We note that the tolerance that doesn't satisfy our protection circuit is when we have a -10% tolerance for the R2 value and a +10% tolerance for our R3 value. We note that this specific tolerance give us a cutoff voltage of 5.69 volts which will cause our battery to get damaged. Furthermore, we note that the +10% tolerance for the R3 value and -10% tolerance for R2 value yields a cutoff voltage to 7.26 volts. This high voltage cutoff will cause our system to run much shorter because a slight voltage drop in our 7.4 volt battery. Ideally, we would like a tolerance of plus or minus 1% because these will only vary our voltage level slightly (6.32-6.48). Although the threshold for our resistance tolerances is plus or minus 5% which gives us a voltage cutoff of 6.02 volts which is .02 volts greater than the voltage cutoff we must enforce. The high end voltage cutoff for this 5% tolerance is 6.81 volts which gives us a bigger voltage difference between out battery and the cutoff. Furthermore, this will still allow us to run our vehicle for a great period amount of time.

6. Cost Analysis

Labor:
Based on [7], a reasonable estimate for the salary (converted to hourly) of an ECE graduate from UIUC is $36 / hour. We estimate that it will take the three of us and average of 15 hours per week to complete this project. Following the equation given in the instructions, the labor cost is then:

$$(\$36 / hr)(2.5)(15\text{hr/week})(12 \text{ weeks}) = \$16{,}200 / \text{person}$$

Then $(\$16{,}200 / \text{person})(3 \text{ people}) = \$48{,}600$ total labor cost.

Part name	Part Number	Unit Cost ($)	Quantity	Total ($)
ESP 8266 Microcontroller	Adafruit Industries LLC #2471	9.95	1	9.95
TL431	Mouser Electronics #595-TL431ILP package TO-92-3	0.44	1	0.44

Part name	Part Number	Unit Cost ($)	Quantity	Total ($)
SN754410	Banana Robotics SN754410	1.75	2	3.50
KA378R05TU	Manufacturer: On semicoductor Digikey part# KA378R05TU-ND	1.25	1	1.25
7.4 Volt LiPo battery 25C 2500 mAh	Amazon Part # B01D9OTRVQ	11.59	1	11.59
Tamiya 70097 motor gear box kit	Pololu item #61	8.95	1	8.95
DC brushed motors	Pololu item #1117	1.79	2	3.58
SFT1350 P-channel MOSFET	Mouser electronics part #863-SFT1350-H	0.92	1	0.92
Passive Components	Resistors and Capcitors	$7.00	1	7.00
TO-220 Heatsink for Voltage Regulator	Jameco Electronics part #326596	0.29	1	0.29
Farhop 4.5 5 mW laser diode module	Amazon part#650-D-9-21-R-5	9.95	1	9.95
HC-SR04 Ultrasonic Range Sensor	Banana Robotics part# BR010020	2.99	3	8.97
Phototransistor Pk wavelength 940nm	LTR-306	0.21	7	1.47
Phototransistor Pk wavelength 630nm	KDT00030TR	0.46	4	1.84
Analog Multiplexer	MUX36S08IPWR	3.57	1	3.57
4 input OR gate	CD4072BM96	0.39	1	0.39

Part name	Part Number	Unit Cost ($)	Quantity	Total ($)
Total Cost	-	-	-	73.66

Table 4: Parts Cost

$$Grand\ Total = Parts + Labor \tag{22}$$

$$Grand\ Total = 73.66 + 48{,}600 \tag{23}$$

$$Grand\ Total = \$48{,}673.66 \tag{24}$$

7. Schedule

Week	Jake	Christian	Martin
2/27	Design review, start research/ prototyping server/client networking	Submit Design Document and start implementation of Motor control circuit on microcontroller	Design Review Finish ordering parts
3/6	continue networking	Finalize motor control implementation and Test	Test analog multiplexer circuit
3/13	start on iPhone app UI, shoot/ toggleMode capabilities	Build LiPo protection circuit and test	Test phototransistor circuits and verify R values for correct outputs.
3/20	Spring Break	Spring break	Begin PCB design

3/27	implement rotation math/ logic	Test Ultrasonic Range sensors to ensure tolerance	Finish PCB design and order
4/3	start experimenting/ developing autonomous mode logic	Implementation of Ultrasonic sensor on microcontroller	Experiment with phototransistor array placement
4/10	continue working on autonomous mode/ obstical avoidance/ target acquisition	Start building physical design of vehicle and optimization of sensor placement	Help with physical design and assembly
4/17	Mock demo, prepare for demo, verify requirements	Finish physical design and ensure the whole system can run for at least 15 minutes. Verify Requirements	Finish assembly Prepare for Demo.
4/24	Demo, work on final paper	Demo and Final Paper	Demo and Final Paper
5/1	Final Paper	Final Paper	Final Paper

8. Ethics and Safety

There are many safety hazards involved with our project but one of the main ones is the use of the 7.4 Volt 2500 mAh 25C LiPO battery. When using this battery, we have to ensure that each cell in this battery is kept above 3.2 volts (6.4 collectively) when discharging it or else the battery could get permanently damaged. Also, we have to ensure that the max voltage is not be above 4.2 volts (8.4 collectively) when fully charged. Furthermore, when charging the battery, we have to ensure that we don’t overcharge or else it could cause a fire. We have to work within these minimum and maximum voltage restrictions to ensure that we use this battery safely. To ensure that these batteries are working properly, we will create a LiPo battery protection circuit for the battery. Also, while using this battery safely, we need to ensure that all our other modules have the correct power, voltage, and current rating to prevent the other components from overheating. As the design process continues, we may need to implement voltage/current readings for certain devices to establish optimal safety and efficiency. Additionally, we take into consideration that our voltage regulator will be dissipating 8.33 watts. This voltage regulator has multiple safety features such as overcurrent protection, thermal shutdown, overvoltage protection, and short-circuit protection. Also, our LiPo battery protection circuit will disable the chip if the battery drops below 6 volts.

As a group of aspiring engineers, we have to abide by the IEEE ethics and ACM guidelines. First and foremost, every project involved has to follow #1 on IEEE code of ethics to ensure the safety of those who will be utilizing this droid which involves power sources and electronic components that could catch on fire or explode [2]. We will also be working with #7 on the IEEE code of ethics to ensure we build the most viable, efficient, and safe project. We will be taking constructive criticism from professors, teaching assistants, and ourselves as group members [2]. We have to keep ourselves honest and ensure that we acknowledge and correct our mistakes. For example, if there is a better way to electronically design our PCB to avoid overheating of components then we have to keep ourselves honest and make this revision. This project entails the aspects of #8 and #10 in IEEE code as well [2]. Whether it is the treatment of group members or other groups, we have to ensure that we don't engage in discrimination of any kind. This will also bode well with #10 in the IEEE code of ethics because it will create a great aura for group members to work in which in turn will enhance the development of our project [2]. We, aspiring engineers, endure #5 on the IEEE code of ethics on a daily basis regardless of whether we are working on a project or not. We have to understand the technology we are using and know the applications/consequences that the project entails. Number 3 on the IEEE code touches in with #5 in the aspect of finding the best application with the available data will enhance the development of our project [2]. We, aspiring engineers, endure #5 on the IEEE code of ethics on a daily basis regardless of whether we are working on a project or not. We have to understand the technology we are using and know the applications/consequences that the project entails. Number 3 on the IEEE code touches in with #5 in the aspect of finding the best application with the available data.

References

[1] Onyx battery manual, Hobbico Inc. Available at: http://manuals.hobbico.com/dtx/dtx-onyx-lipo-manual.pdf

[2] IEEE Code of Ethics, web page. Available at: http://www.ieee.org/about/corporate/governance/p7-8.html Accessed February 2017.

[3] RC Flying Battle Robots, Hammacher Schlemmer, web page. Available at: http://www.hammacher.com/Product/86747?cm_cat=ProductSEM&cm_pla=AdWordsPLA&source=PRODSEM&gclid=CjwKEAiAoOvEBRDD25uyu9Lg9ycSJAD0cnByIKk0xZQLOUp9dgr-qBPlZkX-nBhmWhaOWV50icdiYhoC2PHw_wcB Accessed February 2017.

[4] https://www.nomtek.com/wp-content/uploads/2014/11/8RHfT.png

[5] https://www.ece.illinois.edu/admissions/why-ece/salary-averages.asp