SAFER (Safety Autonomous Following Escort Robot)

Group 16:

Megan White Electrical Engineering

Jordan Harty Electrical Engineering

Jonathan Dillard Computer Engineering

Michael Burton Computer Engineering

Motivation

- SEPS and Knight Ride are great resources to help students cross campus after dark, but may not always be available due to overloaded and school breaks
- Gives students another viable option for them to get home safely at night
- Lights the way home for students walking in dark areas
- Provides surveillance using recorded video of trek
- Acts as a deterrent to possible assailants from approaching students that may be walking alone at night

Goals & Objectives

- Create an autonomous vehicle that will avoid obstacles and follow students to their desired location
- Build a robot that uses Bluetooth capabilities via phone connection to follow the user
- Use computer vision to avoid obstacles in the robot's path
- Provide light to see around the robot
- Record the surroundings as a form of surveillance and possible evidence

Specifications

- The vehicle shall be no larger than 4x4x4 ft
- The vehicle shall be capable of following a person via phone connection or infrared beacon within 15 feet
  - Maintain a steady following distance of 5-7 feet
  - Stop within 2 seconds of user stopping
- The vehicle shall go no faster than 4 mph
- The vehicle shall have a total following distance of at least 1 mile
- The vehicle shall avoid obstacles bigger than a 4x4x4 inch objects
- The vehicle shall be able to store 30 minutes of video
- The vehicle shall be able to establish a Bluetooth connection through a phone app

Standards & Constraints

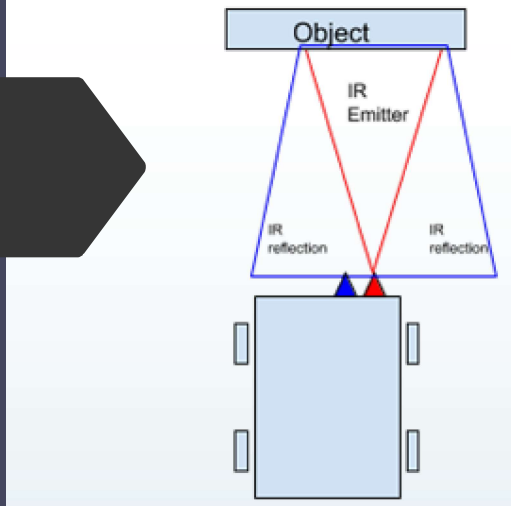
- PCB Standards
- Python Standards
- C Standards
- Bluetooth Standards

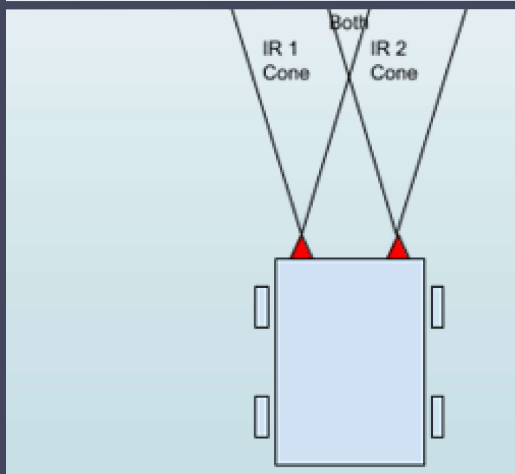
- Environmental Constraint- Weather, avoiding grassy areas
- Health & Safety- Do not want to ram into any people or buildings
- Power Constraint- Robot can only run for so long before needing to be recharged

Robotic Sensing Selection Process for objects & users

Possible IR Configurations

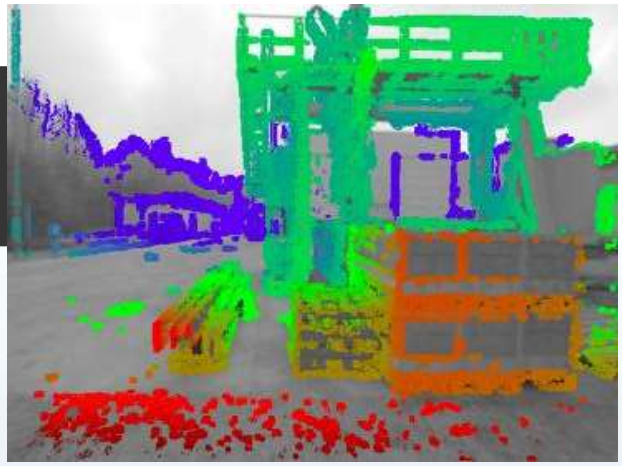
- The top is an example of how SAFER could avoid objects
  - Pros – Cost effective
  - Cons - Lower accuracy
- The bottom is an example of how SAFER follows the user
  - Pros – Not Much
  - Cons - Lower accuracy, extra device needed
- These examples also apply to ultrasonic sensors





Possible CV Configurations

- The top is an example of how SAFER could avoid objects
  - Pros – Accuracy
  - Con – Computationally expensive
- The bottom is an example of how SAFER follows the user
  - Pros – Proven
  - Con- User need to hold pattern

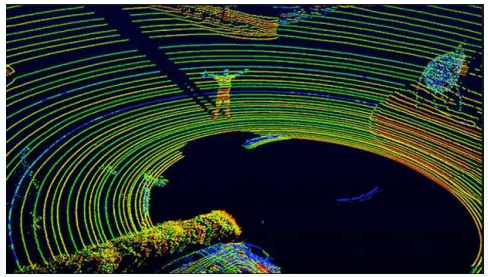


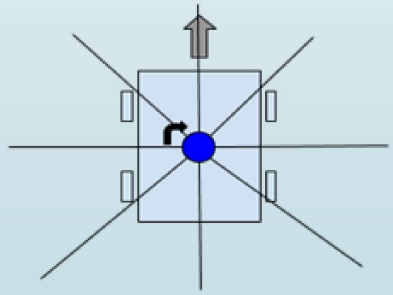
Possible Wireless Signal Strength Configurations

- Follow the user based on the signal strength of the Bluetooth or WIFI connection.
  - Pro – None
  - Con – Very inaccurate

Possible LIDAR Configurations

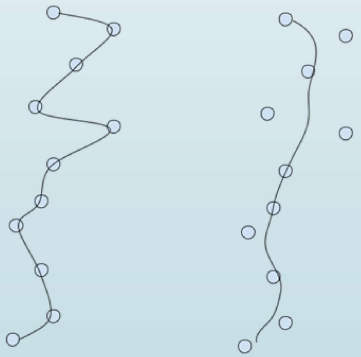
- If we mount a LIDAR sensor to a continually rotating servo we could get a very accurate reading for the distance of object on a 1D plane around SAFER
  - Pros - Gold Standard
  - Con – expensive (possible very expensive)





Possible GPS Configurations

- The user could use an app to connect to the robot and stream their GPS coordinates via Bluetooth
- SAFER will then mimic these coordinates with a GPS and Compass
- To make the robot seem more sober we can average the GPS coordinates so that it results in a smoother path traveled
  - This will hopefully compensate for the lower precision of the GPS
- Pro – Cost, added features with an application
- Con – Possible problems with GPS accuracy



What did we choose?

The SAFER Sensing Stack

Following

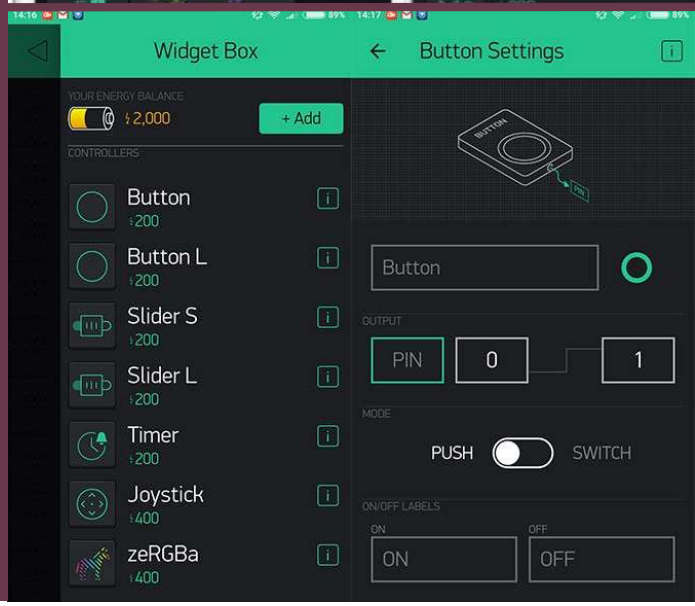
- GPS streaming Via Bluetooth to an Arduino using an app called Blynk

Object Avoidance

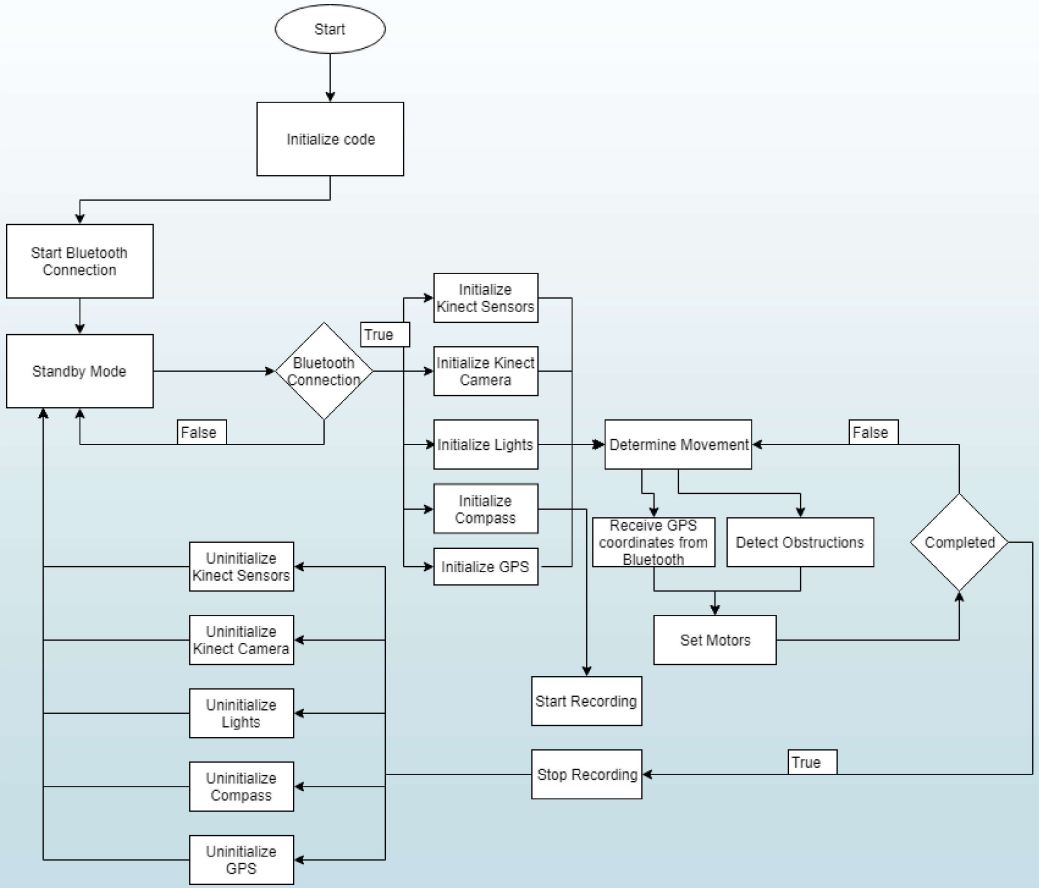
- Computer Vison with a Kinect connected to a Rasbery Pi with Ubuntu 16.04 with ROS (Robot OS)

Mobile Application

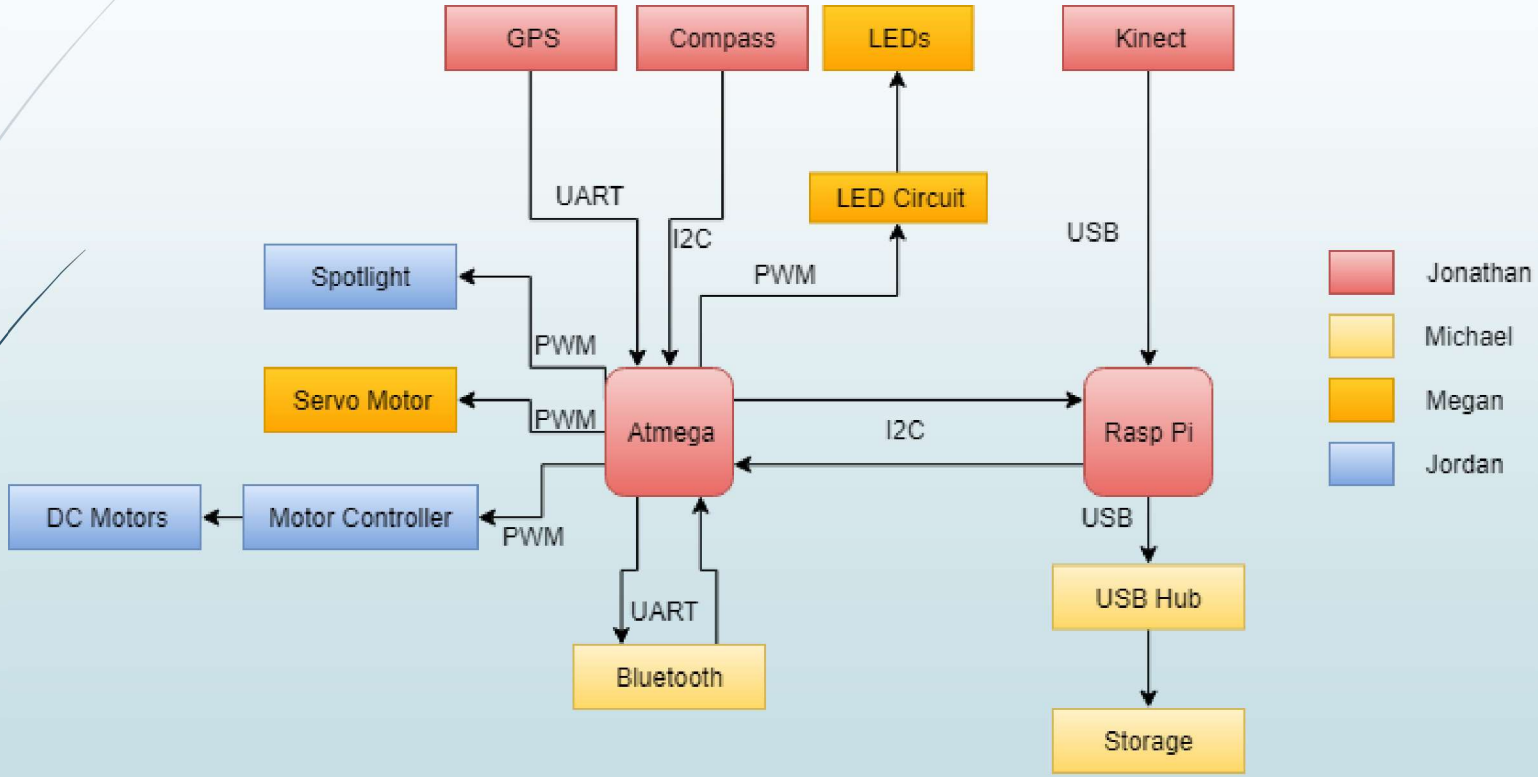
- We planned on making out own app to control SAFER but during our research we found Blynk
- Blynk allows us to build custom interfaces to control the robot and connects via out Bluetooth module



Software Design



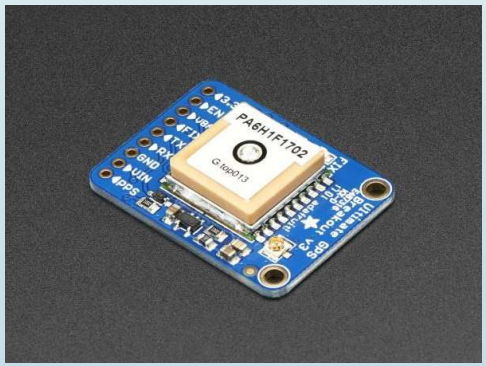
Block Diagram



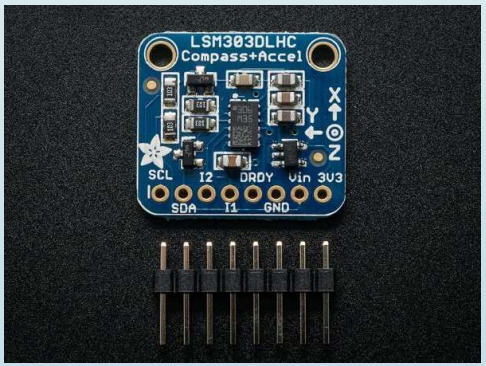
Part Selection

GPS Module / Compass Module

- Adafruit Ultimate GPS Breakout
  - Runs on 5V and 20mA
  - Highly Sensitive Receiver
  - Enable/Disable functionality
  - Software Libraries



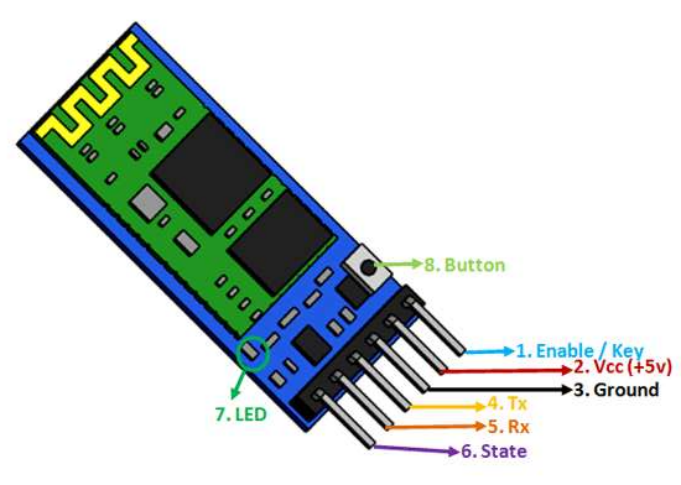
- LSM303DLHC Compass
  - Runs on 3.3V or 5V
  - I2C communication
  - Built accelerometer
  - Software Libraries



Microsoft Kinect

- Offers Multiple Sensors
  - RGB Camera
  - Depth Sensor
    - Infrared Grid and detection
  - Microphone Array
- Open source libraries
- ROS Integration

Bluetooth module HC-05



- Most common module used for Arduino
- Master and Slave on one chip
- V2.0 Searl port profile (SPP) protocol standard
  - Great for sending bursts of data
- 3.6 - 6 Volts

Microcontroller & Raspberry Pi

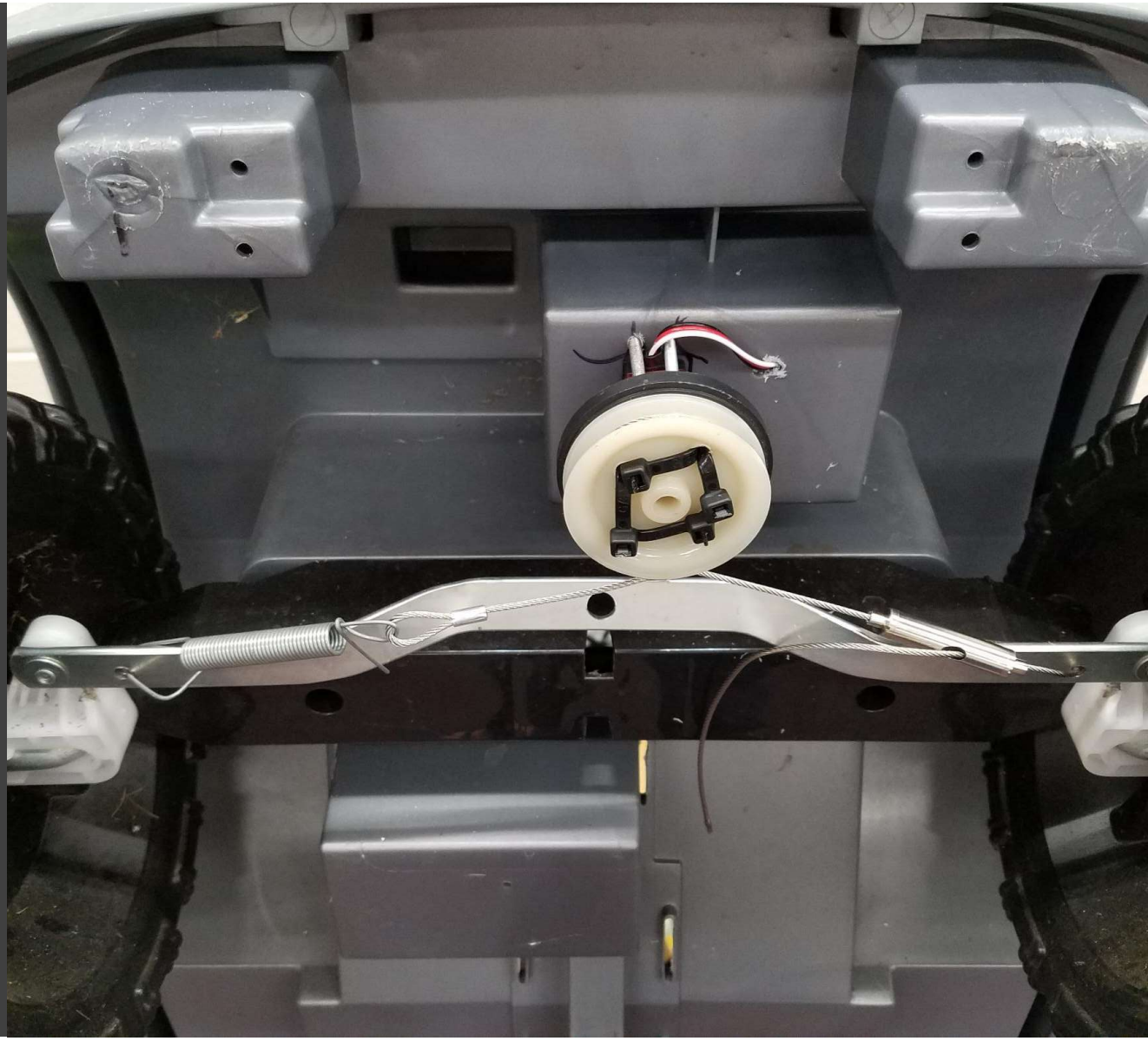
- For the custom PCB, an ATmega328P will be used to control lights, steering, driving, and Bluetooth
- The ATmega328P was chosen because they are the microcontrollers used on Arduinos, which are easy to code, test, and use
- To handle computer vision, however, a Raspberry Pi will be used
- The Raspberry Pi will connect to the Microsoft Kinect and send all necessary information over to the MCU on the custom PCB to help make decisions

Chosen Vehicle: Power Wheels

- Since building a base would be timely and possibly costly it was in our best interest to purchase a vehicle that is cheap, functioning, and small enough to get around campus
- A Power Wheels was chosen, a toy meant for small children to be able to emulate driving a car
- Fairly common, these vehicles can only be used for a short time before the child outgrows them, so plenty of used ones can be found online for relatively cheap

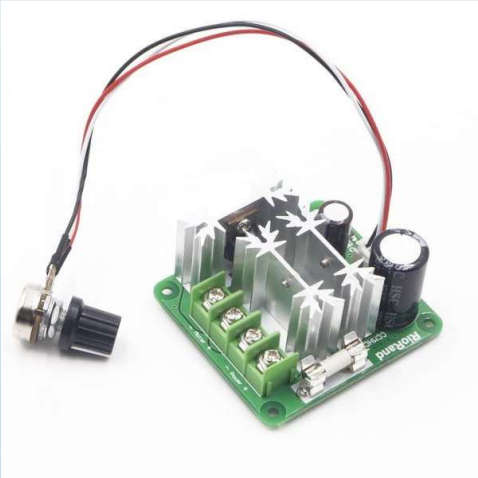
Autonomous Steering

- Power Wheels are usually driven, so to make the vehicle autonomous a system had to be devised to allow the wheels to turn without a driver
- Shown is our current solution, which uses a servo to rotate a wire left or right to turn the wheels



DC Motor Comparisons

Model	Operating Voltage (V)	Max Current (A)	Rated Current (A)	Price ($)
RioRand	6-90	15	8	12.99
SMAKN	10-40	15	10	12.00



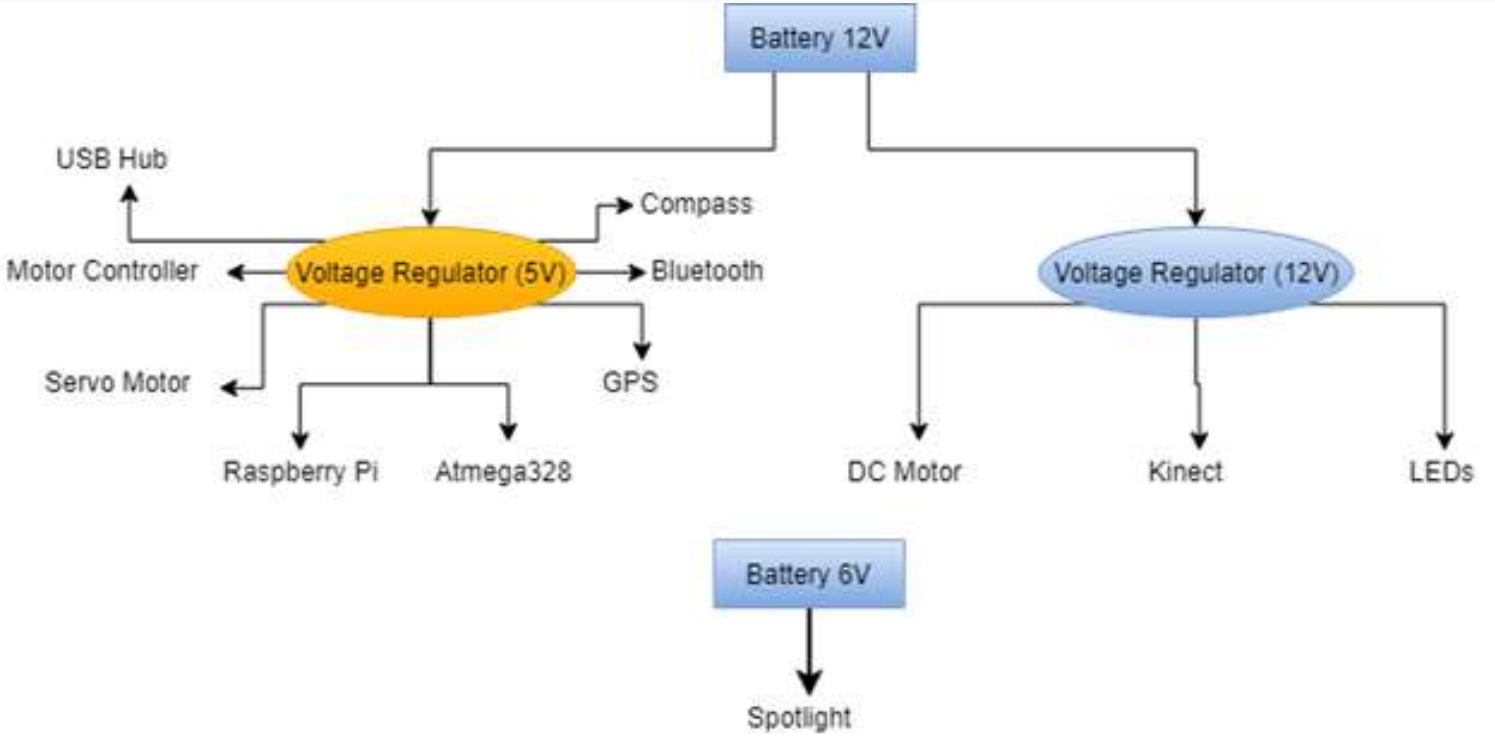
LED Strips Comparison

Model	Lumens per Foot	Power (W)	Length (in)	Price ($)
LED-T2430L-1-WT	135	2	12	19.00
LED-T24W-1-WT	200	3	12	27.00
Kichler 6HS30K12AL	215	4	12	16.13
WAC Lighting LED-T24C-2IN-WT	200	0.5	2	7.50
Kichler 6HS30K06AL	215	2.7	6	11.25
Progress Lighting P7040-30	120	3	12	24.39
WAC Lighting LED-TX2430-6IN-WT	275	2	6	21.00
HitLights SMD3528	72	1.3	16.4	7.99

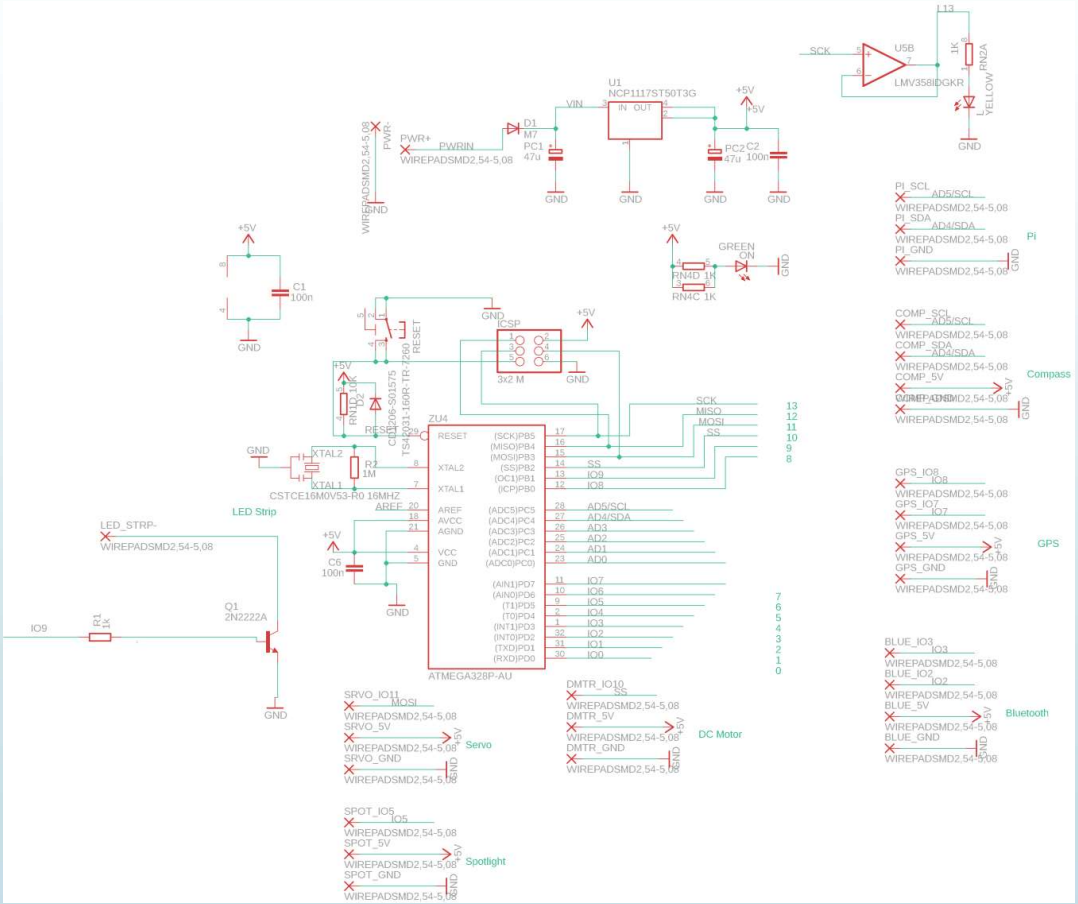
LED Headlights

- 6000 lumens
- Charge time: 8 hours
- Run time: 16 hours
- Current: 9 A

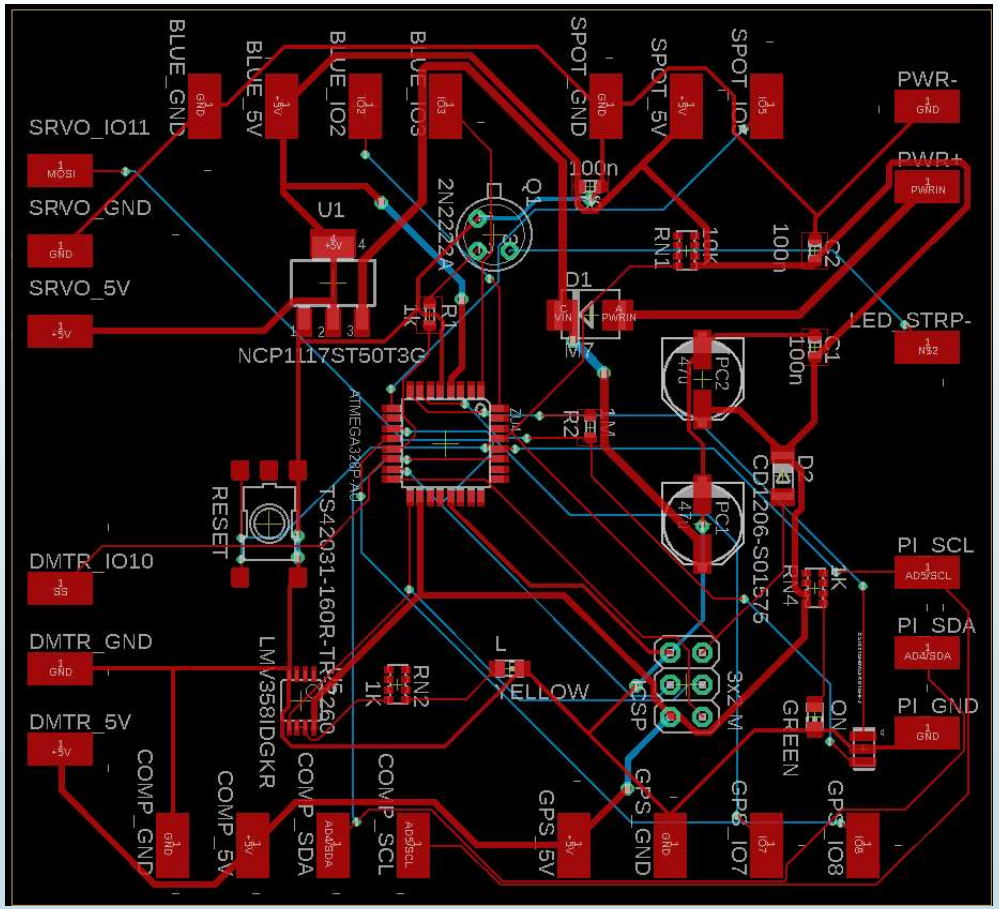
Power Management



PCB Schematic



PCB Layout


BLUE_GND
BLUE_5V
BLUE_IO2
BLUE_IO3
SPOT_GND
SPOT_5V
SPOT_IO5
PWR-
PWR+
SRVO_IO11
SRVO_GND
SRVO_5V
U1
NCP1117ST50T3G
2N2222A
Q1
100n
RN1
10K
100n
Q2
D1
R1
1k
LED_STRP-
Q1
100n
PC2
47U
R2
1M
ATMEGA328P-AU
D2
CD1206-S01575
PC1
47U
RESET
TS42031-160R-TR-260
DMTR_IO10
DMTR_GND
DMTR_5V
LM358IDGKR
RN2
1K
L
YELLOW
RN4
5K
PI_SCL
PI_SDA
PI_GND
3X2_M
DSP
ON
GREEN
COMP_GND
COMP_5V
COMP_SDA
COMP_SCL
GPS_5V
GPS_GND
GPS_IO7
GPS_IO8

Administrative Content

Work Distribution

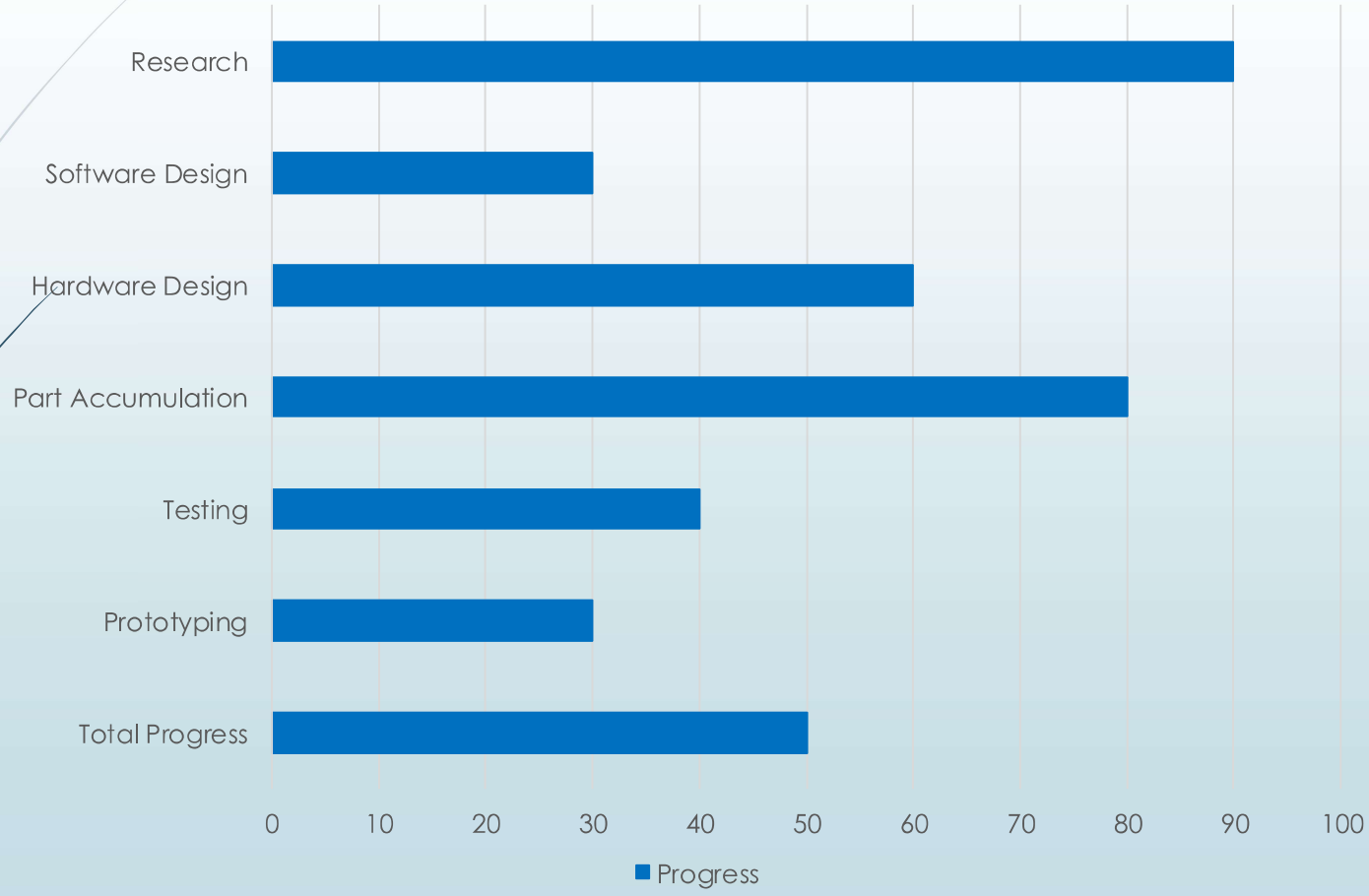
	Power	Computer Vision	Lighting	PCB Design	Steering & Drive	Mobile App
Megan	2		2	2	1	
Michael		2			2	1
Jonathan		1		1		2
Jordan	1		1			

1- Primary
2- Secondary

Budget

Item	Price/Unit	Quantity	Total Cost
Power Wheels	~$200-$400	2	$25
Power Wheels Replacement Battery 6V	$35	1	$35
Microsoft Kinect	$40	1	$0
Raspberry Pi 3 Model B	$35	1	$0
G-Drive 1 TB Hard Drive	$70	1	$0
Triple-axis Accelerometer & Magnetometer Compass Board	$14.95	1	$14.95
GPS 10 Hz Modules	$39.95	1	$39.95
8 Ohm Loudspeaker	$17.95	1	$17.95
Rechargeable Searchlight LED Flashlight 6000 Lumens	$28.68	2	$57.36
16.4 Ft 12 V DC LED Strip Lights	$10.68	1	$10.68
HC-05 Bluetooth Module	$11.11	1	$11.11
DC Motors	$6.95	2	$13.90
DC Motor Controllers	$12.95	2	$25.90
Total			$251.80

Progress


Research
Software Design
Hardware Design
Part Accumulation
Testing
Prototyping
Total Progress
0
10
20
30
40
50
60
70
80
90
100
Progress

Issues

- Coding for the steering servo motor
- Finding an appropriate dc motor/ESC
- Integrating information from sensors to hardware

Next Steps

- Hook up new DC motors and test steering and driving
- Order and test initial PCB design
- Connect the Microsoft Kinect and write code for object detection

Questions?