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 when they feel unsafe, but may not always be available due to being too overloaded as well as school breaks
- Gives students another viable option for them to get home safely
- Lights the way home for students walking in dark areas
- Provides a place to put a backpack
- Allows student to start an Emergency Mode to draw attention to an emergency situation
- Provides us with a great way to learn about areas that we are interested in, such as computer vision and autonomous driving

Goals & Objectives

- Create an autonomous vehicle that will follow students to their desired location
- Implement gradual autonomous steering and driving
- Build a robot that uses Bluetooth capabilities via phone connection to communicate with the user
- Create a mobile application
- Provide light to see around the robot
- Provide audio amplification for a speaker
- Room on the robot to place a bag
- As a stretch goal, create an Emergency Mode that will play a siren and flash lights to alert nearby people
- As a stretch goal, create a Remote Control Mode that will allow the user to use the phone app to move the robot

Requirement Specifications

	Max:
Vehicle dimensions	4x4x4 ft
Following distance	7 ft
Stopping distance	1 ft
Travel distance	1 mile
Speed	4 mph

Part Selection

Chosen Vehicle: Power Wheels

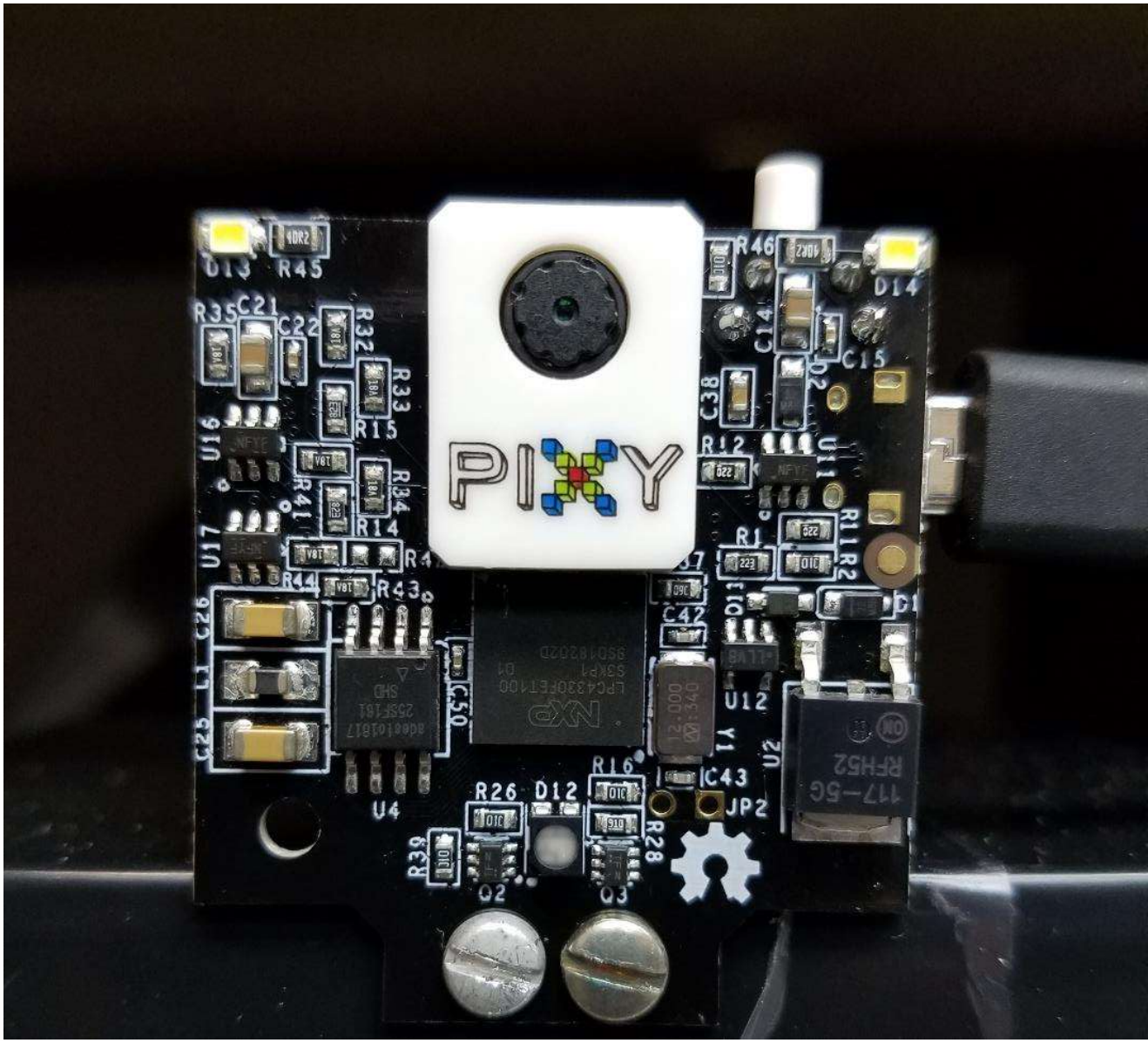
- Since building a base would be timely and possibly costly it was in our best interest to purchase a vehicle that is cheap, functioning, and small enough to get around campus
- A Power Wheels was chosen, a toy meant for small children to be able to emulate driving a car
- Fairly common, these vehicles 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

Microcontroller & Raspberry Pi

- For the custom PCB, an ATmega328P is 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 PixyCam and send all necessary information over to the MCU on the custom PCB to help make decisions

PixyCam

- The Pixy2 is a useful & fast vision sensor that is made for computer vision
- It comes ready out of the box able to sense objects by their colors or bar codes
- Using PixyCam software the camera can be calibrated as needed and then connected to the Raspberry Pi, which will be used to handle the video processing



Linear Potentiometer

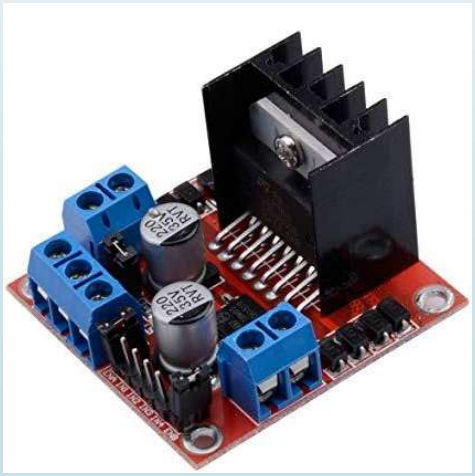
- Bourns PTA6043-2015CPB103
- 10 kΩ Sliding Linear Potentiometer
- Three inputs: 5 V, Ground, and Signal

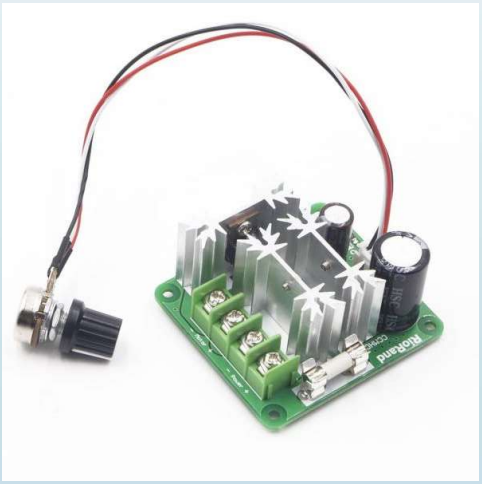
DC Motors

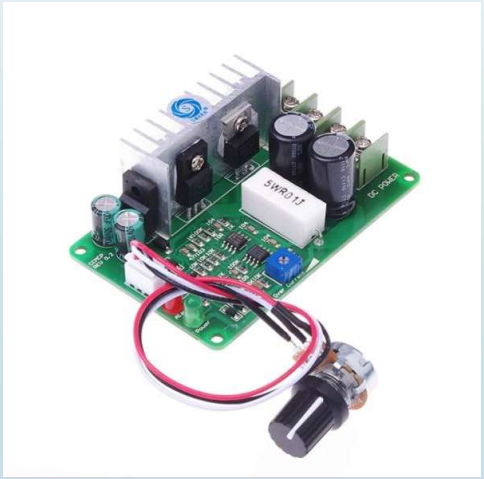
- 6-24 Volt DC High Speed Hobby Motor
- Gear Head DC Motor 12Vdc

DC Motor Comparisons & H-Bridges

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
L298_H_Bridge	46 (max)	4	2	6.99







Speaker

Speaker	Supplier	Power Supply?	Sound Quality	Price
Speaker for Raspberry Pi	Dexter Industries	Yes	Okay	$7.99
Z50 Grey	Logitech	No	Good	$16.99
Docooler MiniHamburg Speaker	Docooler	Yes	Poor	$6.36
Mini Portable Speaker Plug & Play	Leadsound	Yes	Good	$16.99
DROK 15W Speaker	DROK	No	Best	$17.95

LED Headlights

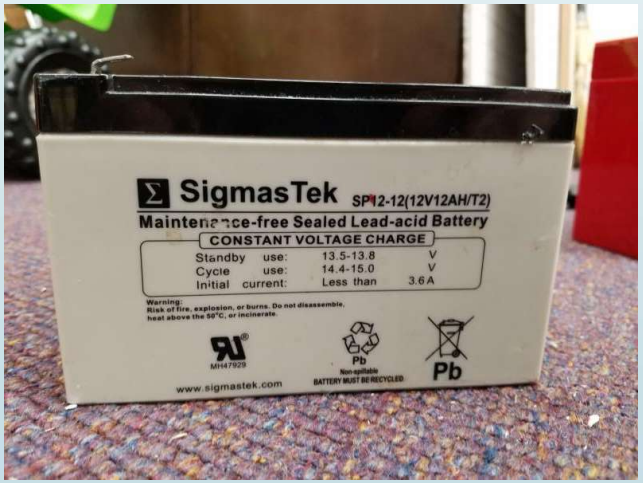
- 6000 lumens
- Charge time: 8 hours
- Run time: 16 hours
- Current: 9 A

LED Strips Comparison

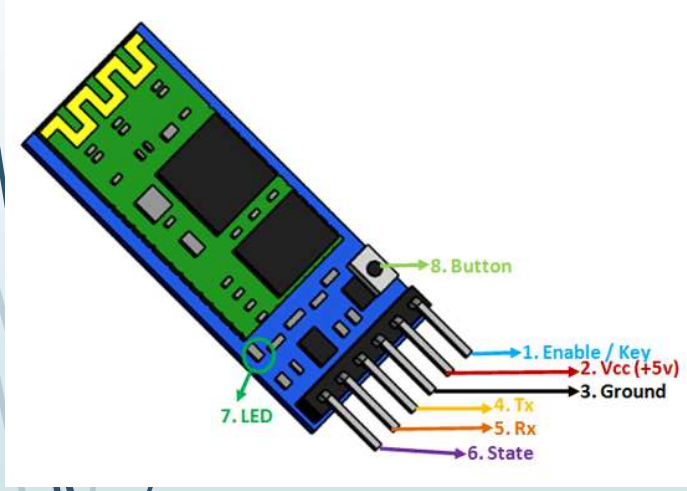
Model	Lumens per Foot	Power (W)	Length (in)	Price ($)
LED-T2430L-1-WT	135	2	12	19.00
LED-T24W-1-WT	200	3	12	27.00
Kichler 6HS30K12AL	215	4	12	16.13
WAC Lighting LED-T24C-2IN-WT	200	0.5	2	7.50
Kichler 6HS30K06AL	215	2.7	6	11.25
Progress Lighting P7040-30	120	3	12	24.39
WAC Lighting LED-TX2430-6IN-WT	275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

Batteries

- The red 6 volt battery (left) was replaced by two of the 12 volt batteries (below).



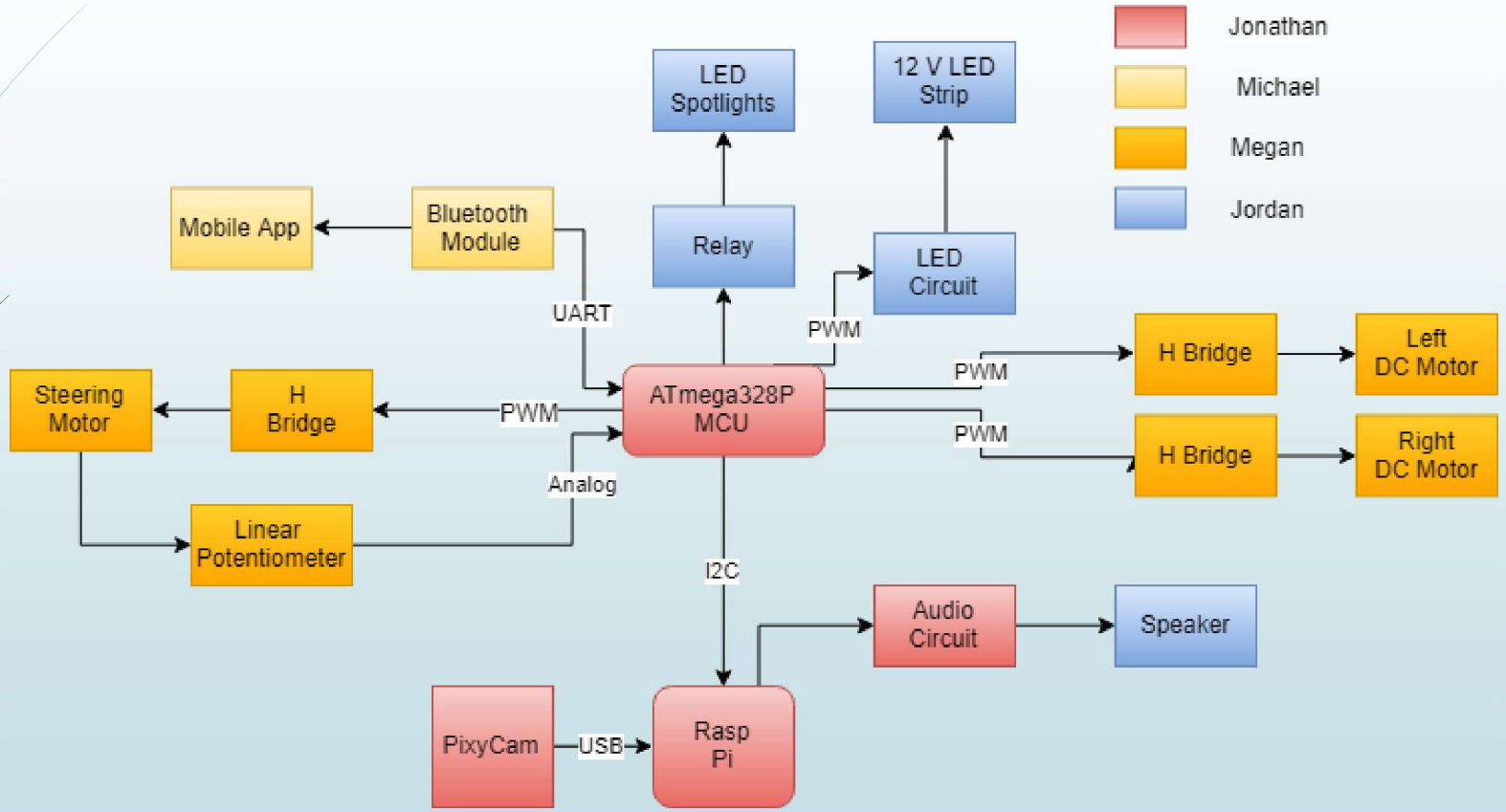
Bluetooth module HC-05



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

Hardware

Block Diagram

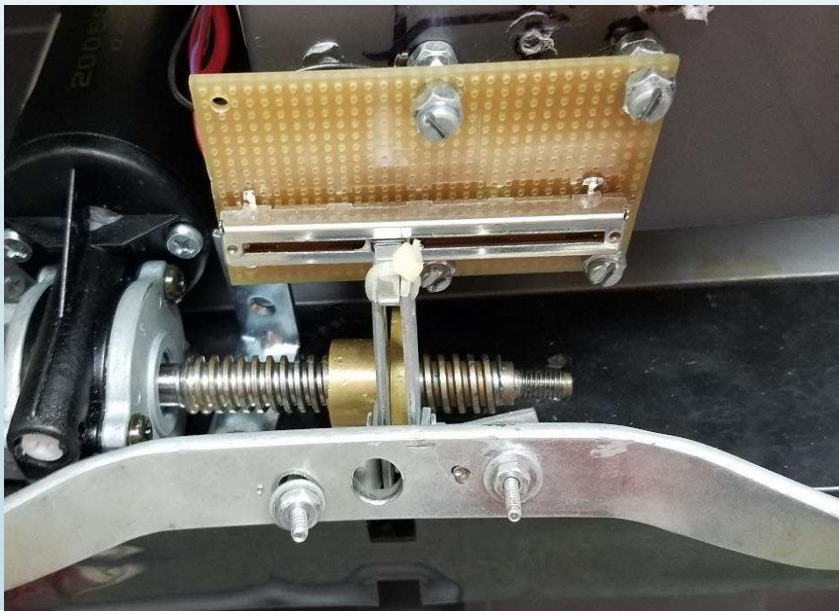


Autonomous Steering

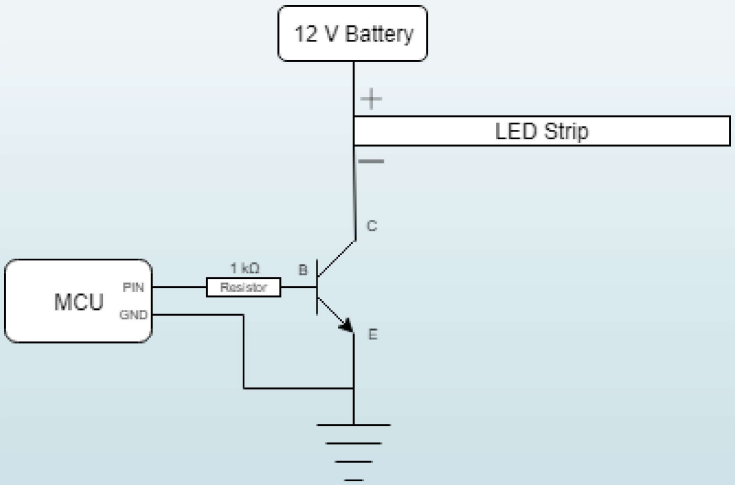
- Power Wheels are usually driven, so to make the vehicle autonomous a system had to be devised to allow the wheels to turn without a human driver
- Shown is our solution, which uses an H-Bridge to control the polarity of the shown DC motor with a 3" axle that rotates CW or CCW, which in turn moves the gear linearly
- The axle of the car is mounted to the gear, so that as the gear moves the axle is forced left or right
- I

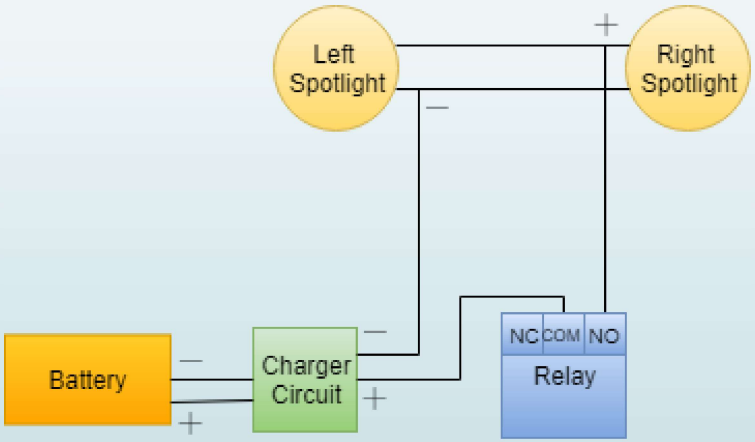
Linear Potentiometer

- In addition, a 10 kΩ sliding linear potentiometer is mounted in order to give feedback about the gear's position to the MCU
- This is mounted to the bottom of the vehicle to track where the moving gear is. The potentiometer sends a signal back to the microcontroller on the custom PCB between the values of 0 to 1023, and the code knows which values mean that the wheels are turned right, left, and center



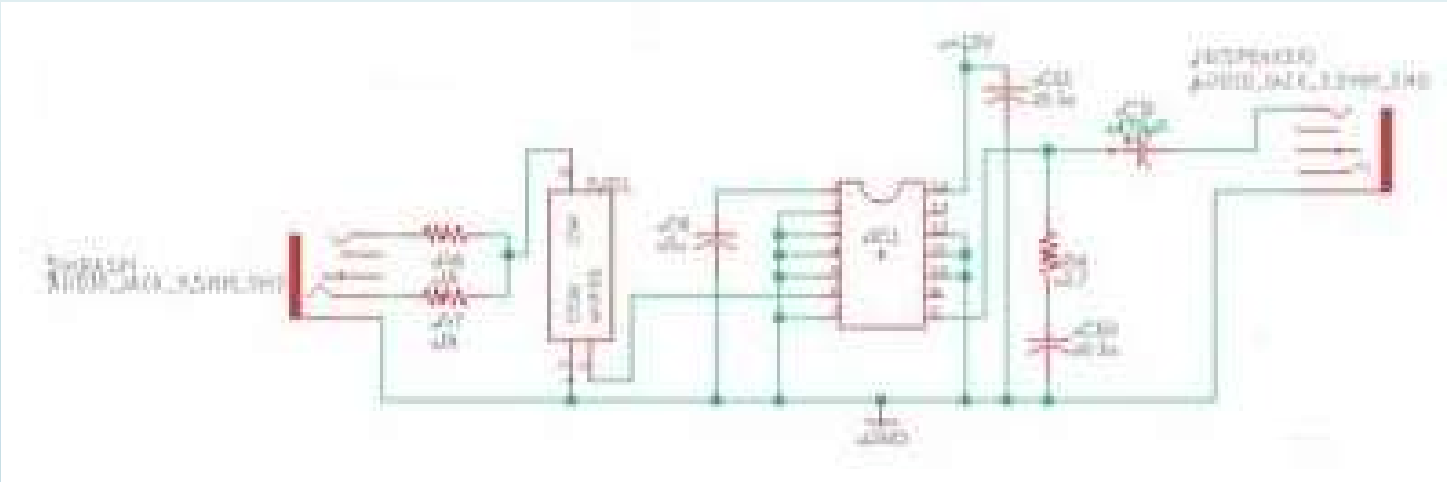
LED Light Strips & Headlights



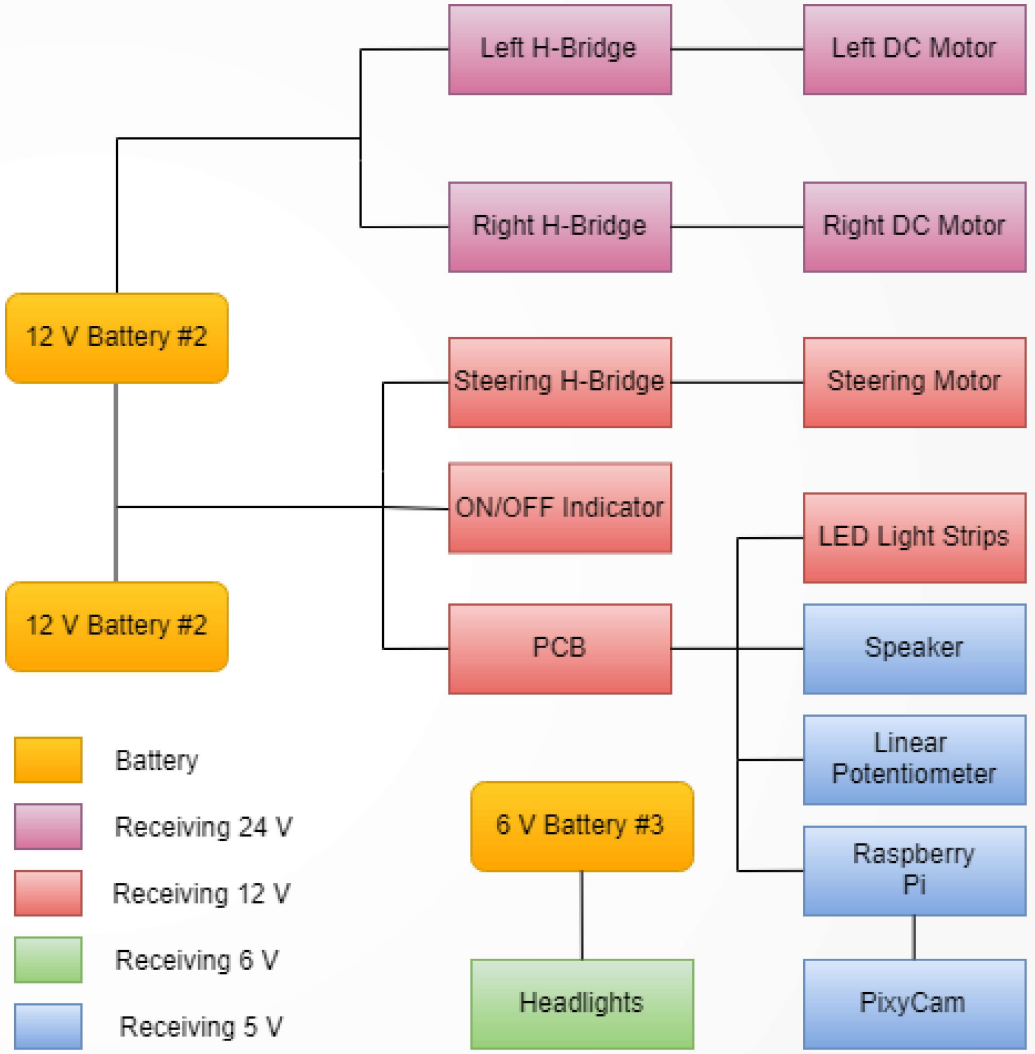


Speaker

- Pi stores the audio files
- Holes to increase audio quality
- Audio amplifier circuit

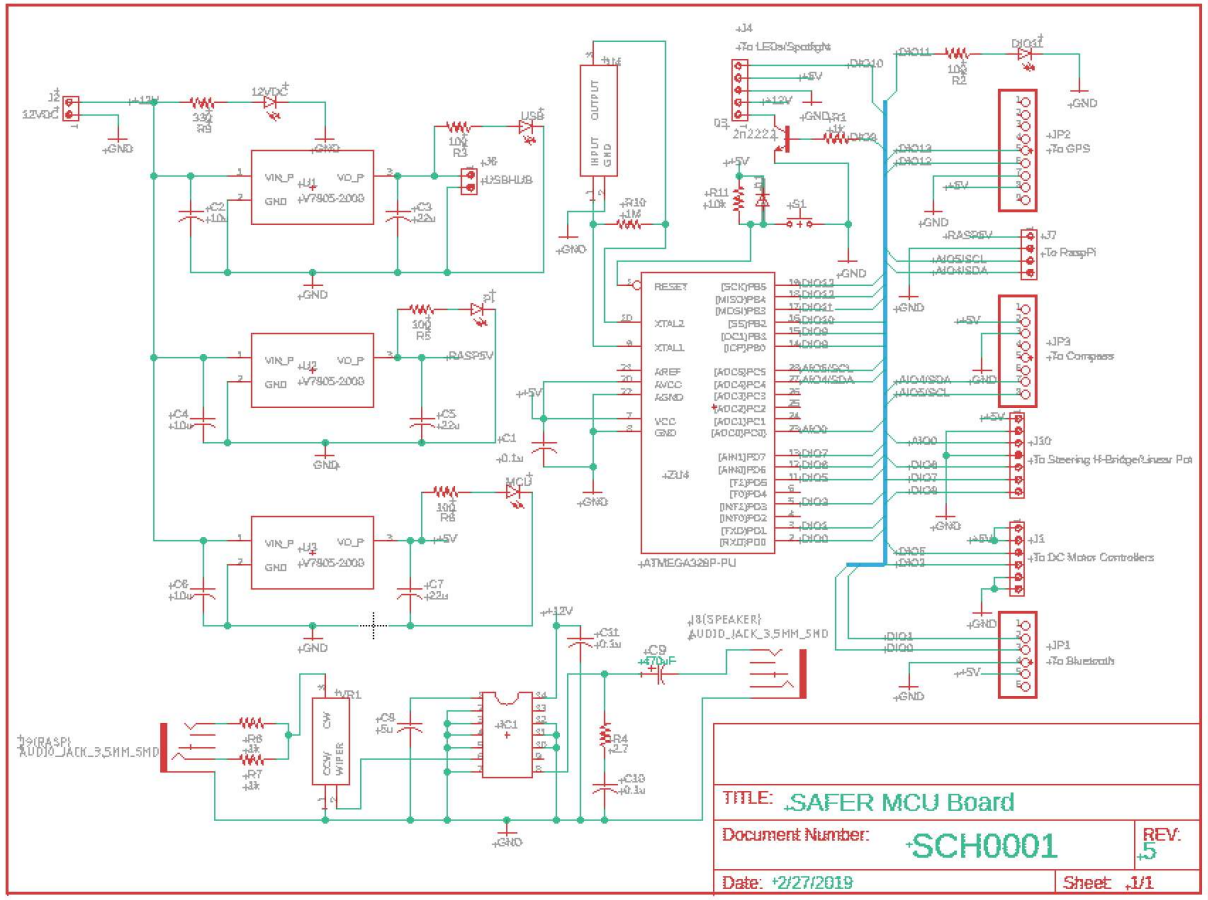


Power Management

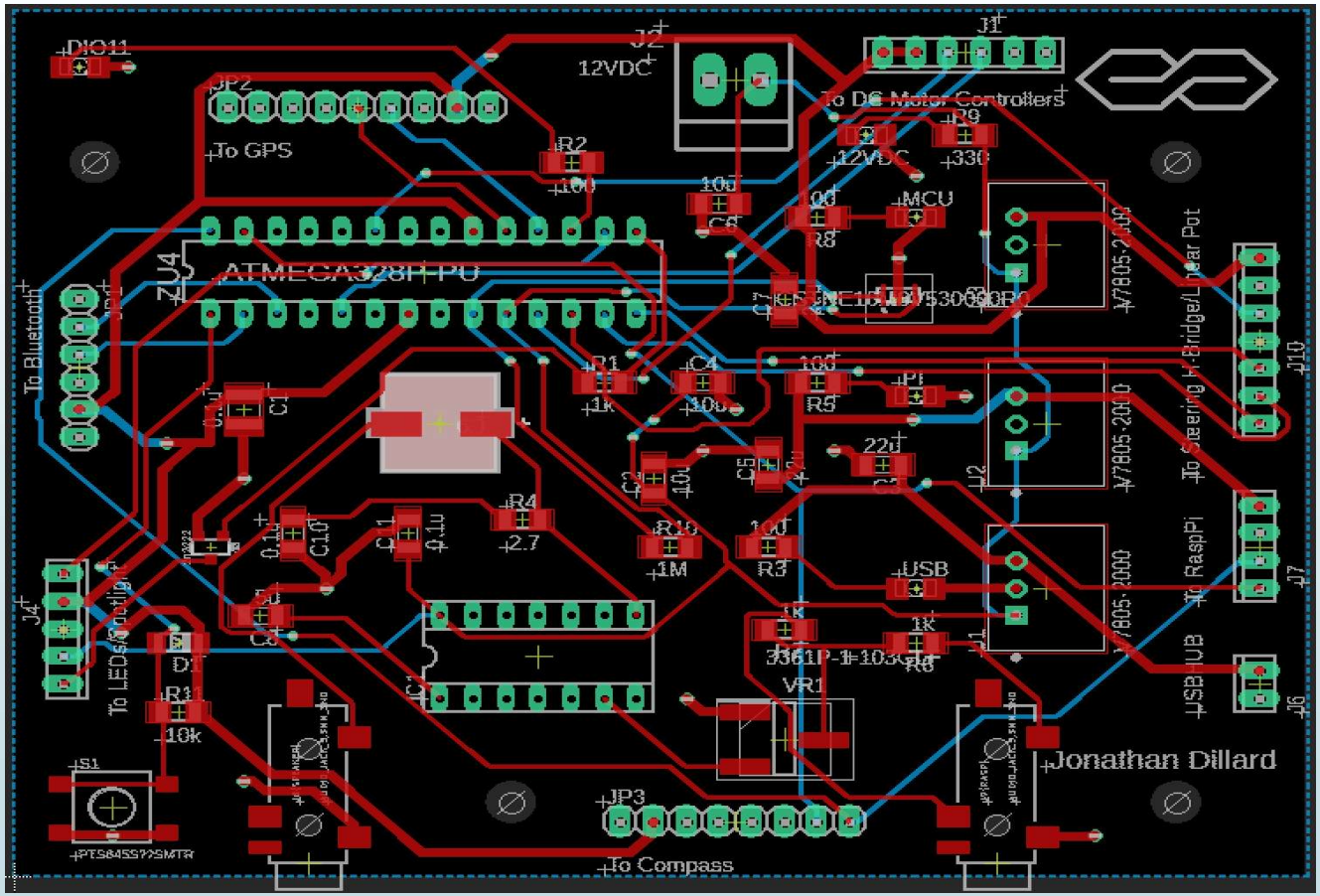


PCB Schematic

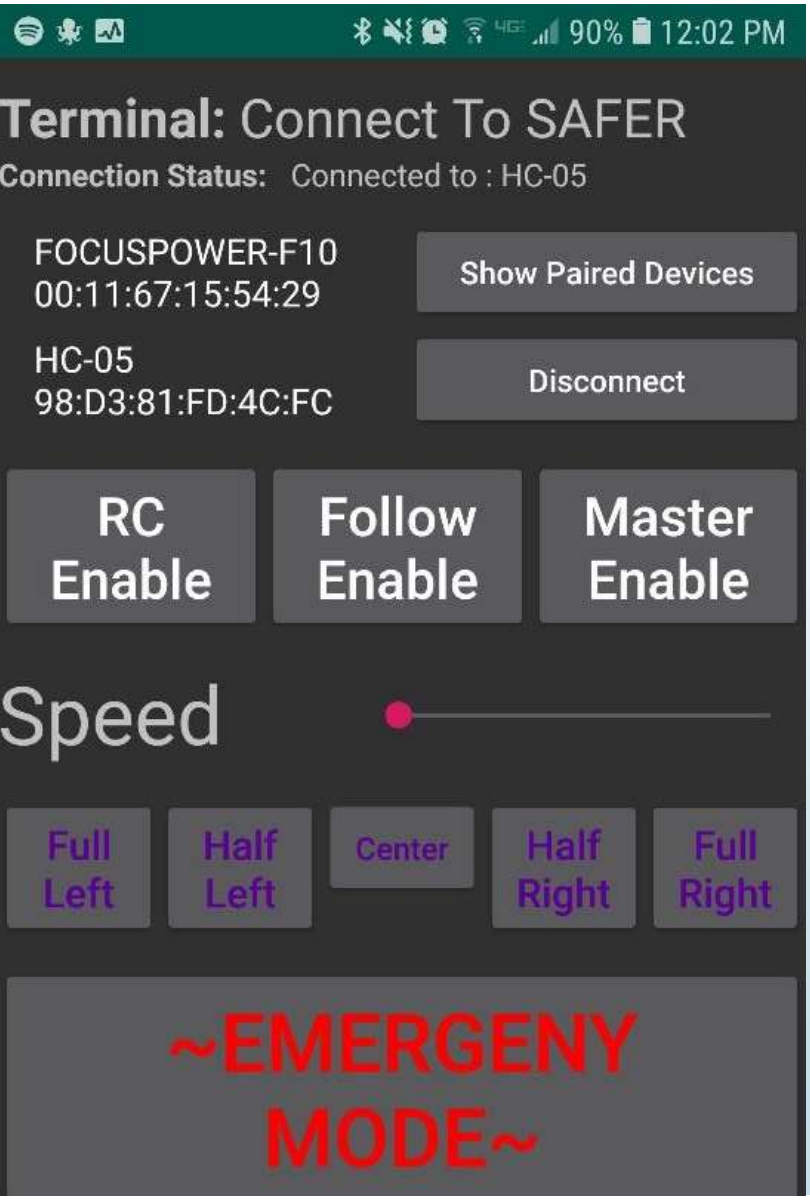
- MCU
- Audio Amplifier Circuit
- Voltage Regulators
  - Raspberry Pi 5V 2A
  - MCU and peripherals 5V 1A
  - Possible USB Hub 5V 2A
- LED circuit
- Reset
- Pin connectors



PCB Layout


12VDC
To GPS
To DC Motor Controllers
ATMEGA328P-PU
To Bluetooth
To Steering H-Bridge/Linear Pot
To RasPi
To LEDs/Switch
To Compass
Jonathan Dillard

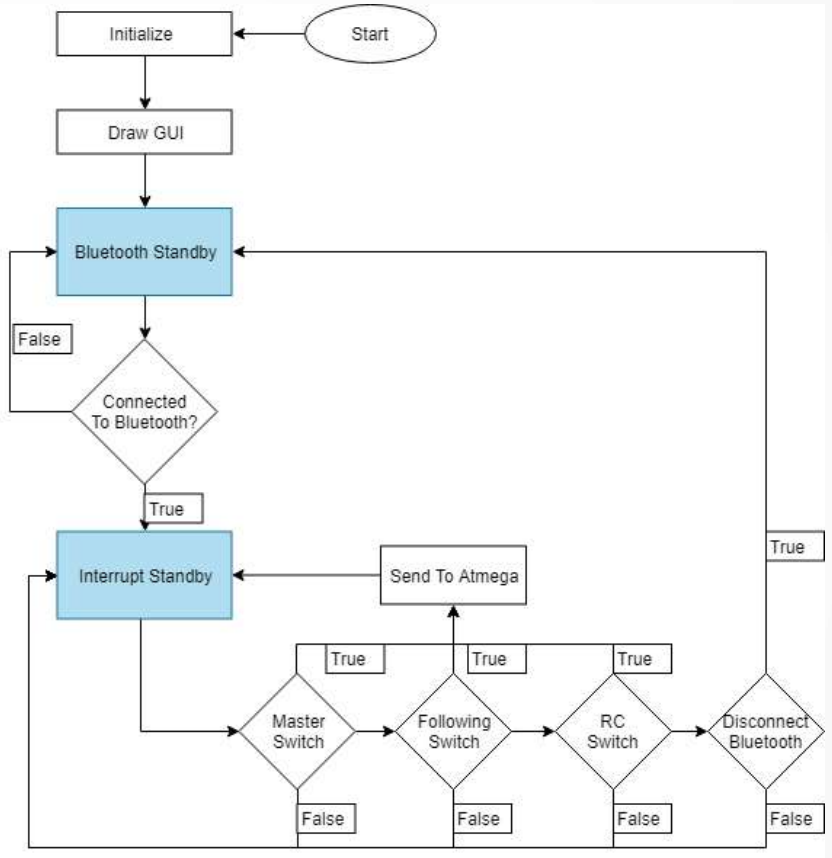
Software



Mobile Application

- We designed our own Android Application to pair with and control SAFER
- The user can:
  - See what the SAFER robot is telling them
  - Enable/Disable the robot
  - Remote Control the SAFER robot
  - Enable follow mode
  - Enable emergency mode

Phone App Flow Chart



ATmega328 Software

The ATmega328 Is the control center of SAFER

- ATmega328 is responsible for
  - Bluetooth connection to phone app
  - I2C connection to Raspberry Pi
  - Controlling Steering motor and linear potentiometer
    - Utilizing Interrupts
  - Controlling two back driving motors
  - Controlling the headlights
  - Controlling the LED strips for ground illumination

ATmega328 Connections

- Bluetooth Connection
  - Wireless up to around 10 meters
  - Serial Communication at 9600 baud
  - Used for two-way communication
    - AT -> APP : Sends status strings to communicate with the user.
    - APP -> AT : Sends enable, follow, and RC commands.
- Inter-Integrated Circuit (I2C) Connection
  - Wired Connection
  - Used for two-way communication
    - PI (Master)-> AT : Sends speed and steering information calculated from PixyCam data.
    - AT (Slave)-> PI : Sends enable and disable flags to play sound for emergency and other modes when requested.
  - Ran at a speed of 100 kHz

ATmega328 Motor Control

- Front Steering
- Analog Connection to a linear potentiometer to read the positions of the wheels
- PWM connection to an H Bridge to control the direction and speed of the steering motor.
- Back Wheels
- PWM connection to control the speed of the motors.

Steering Timer Interrupt

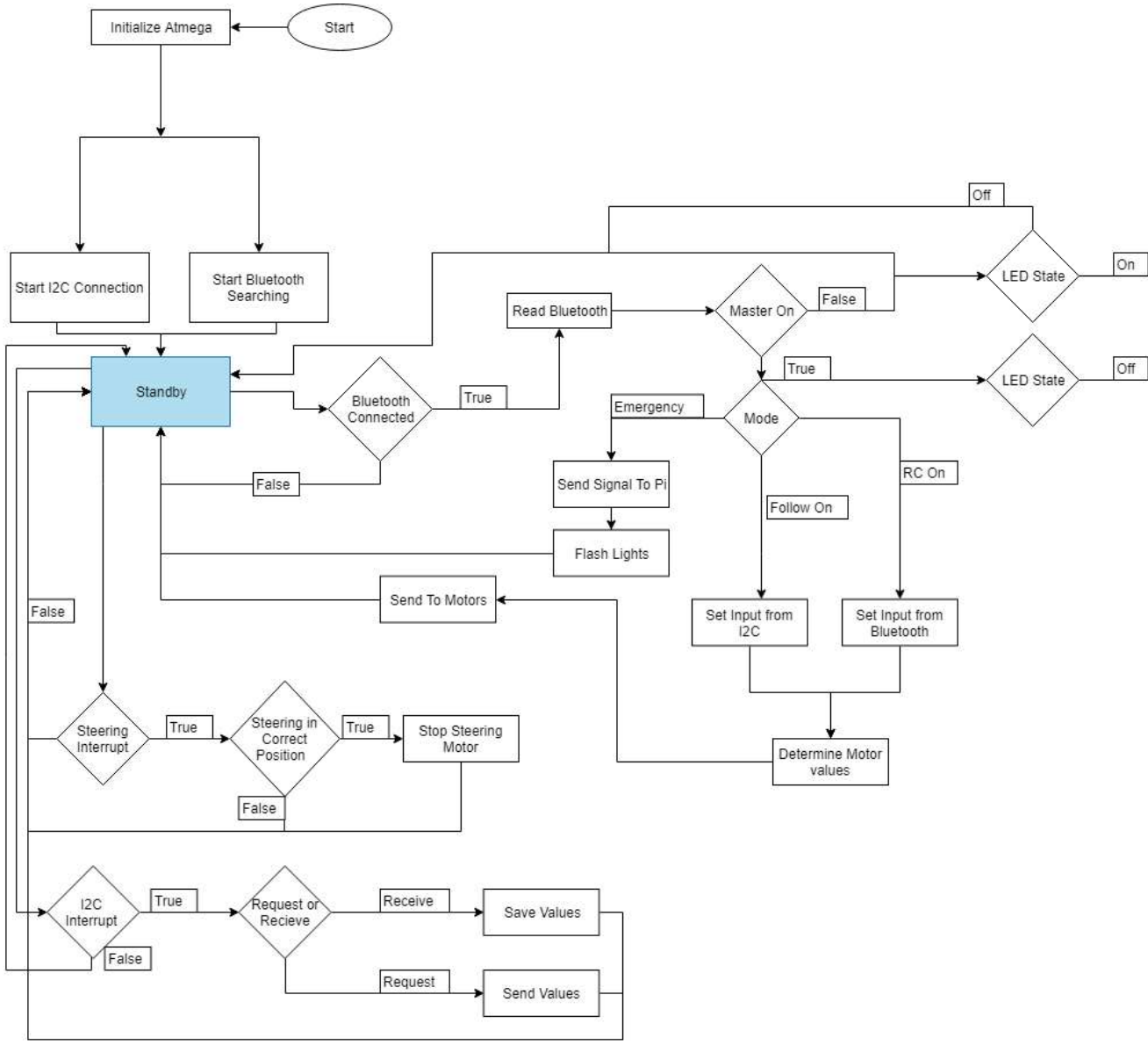
- Original Code:
- A function was called that looked like this:
  - Int steeringPos;
  - While (steeringPos != target) {
    - SteeringPos = readValue();
  - }
- This would block another code from running.

- With Interrupt:
- A function is called to set a global target and start steering
- Every 50 ms or so an interrupt is called to check if the motor should stop.
- This lets other code run maintain a real-time system.

ATmega328 Light Control

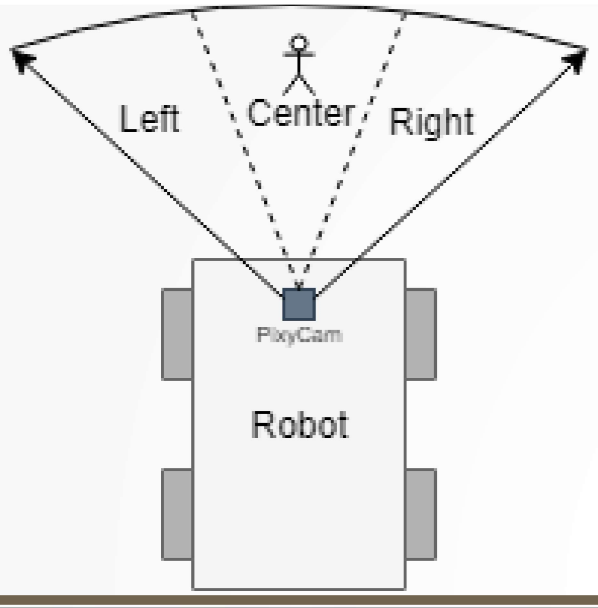
- Head Lights
  - Utilizes 5V output signal
  - The software alternates the state of the signal pin when necessary.
  - System Enabled
    - Set output pin to a constant high
  - Emergency Mode
    - Alternate the state of the pin
  - System Disabled
    - Set output pin to low
- LED Strips
  - Utilizes a PWM signal to control LED strip intensity
  - Software sets a full 5V PWM signal.
  - System Enabled
    - Set output signal to full 5V PWM signal
  - Emergency Mode
    - Alternate the state of the signal between 5V and 0V PWM signals
  - System Disabled
    - Set the output signal to a 0V PWM signal

ATmega328 Flow Chart



Raspberry Pi Software

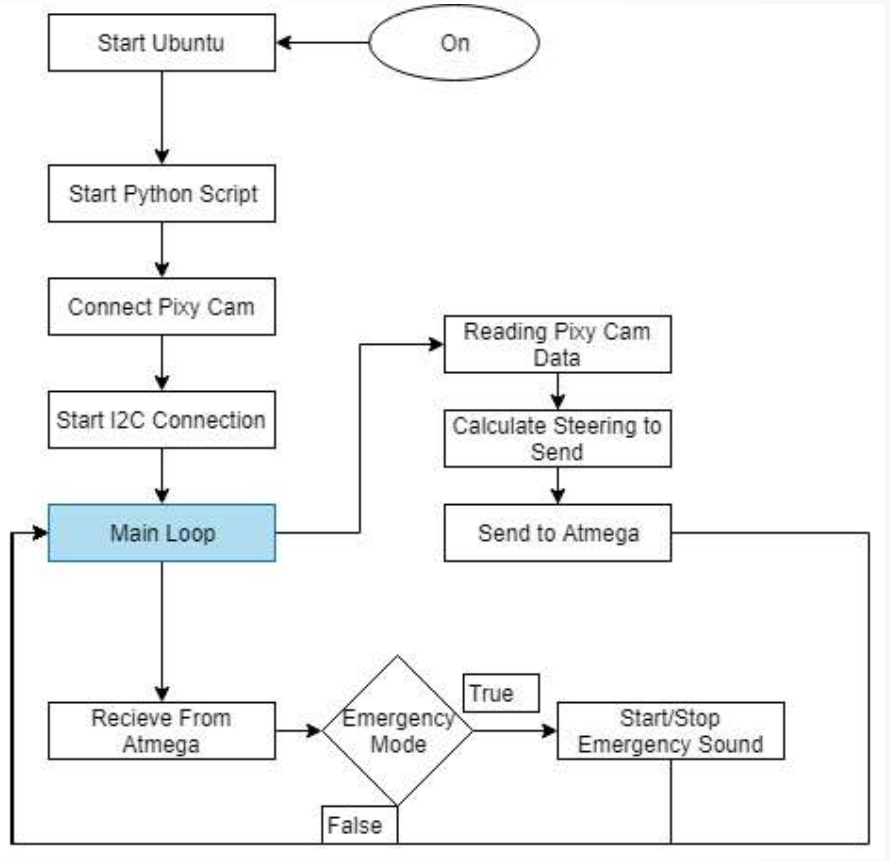
- Request flags from the Arduino
  - Flags enable/disable modes on Raspberry Pi
- Analyzes data received from the PixyCam
  - Determines appropriate speed and direction based on coordinate and dimension information.
  - Send speed and direction information to Arduino
- Create and control Audio threads during the use of different modes.



How to calculate the following vectors (Computer Vison)

- Speed
  - The needed speed is calculated by measuring the size of the object in the frame in pixel
  - Through data driven testing we have an established formal to real MAX(height, width) to a speed to pass to the ATmega
- Steering
  - Based on the X coordinate of the bottom right corner of the PixyCam's FOV which sends a value to the ATmega setting its new steering target

Rasbery Pi Flow Chart



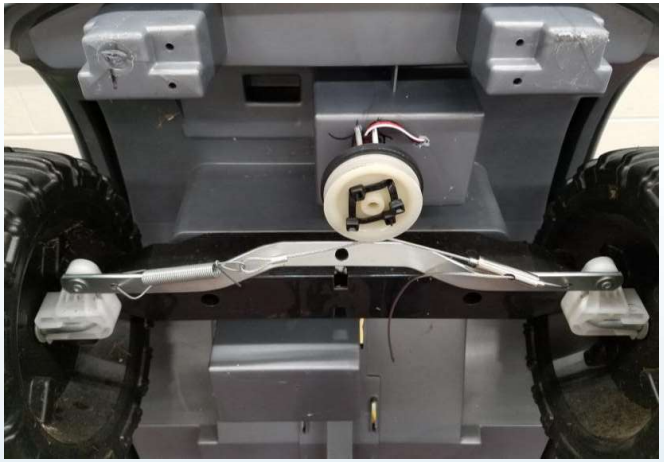
Challenges

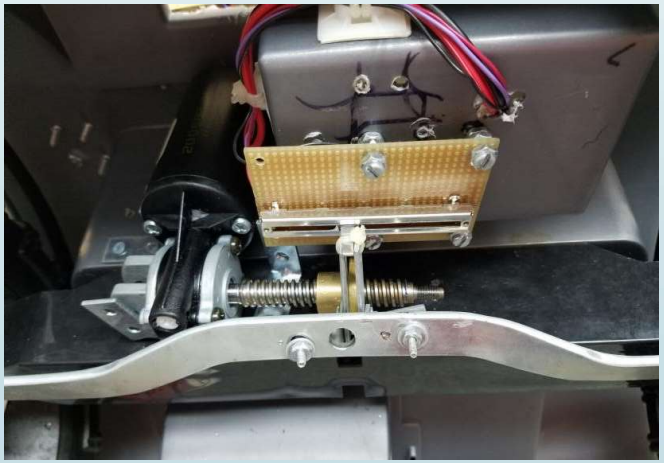
- Autonomous Steering
- PWM Compatibility
- Vehicle Speed
- Blynk App "Packet Too Big"
- GPS and Compass Inaccuracy

Issue: Autonomous Steering

- Weight and insufficient torque produced from the pulley system didn't allow the wheels to turn.

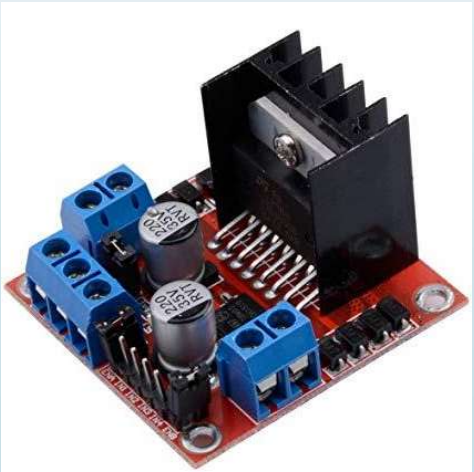
- Solution: 12 Volt DC Gearhead motor and a nut to turn circular motion into linear motion.





Issue: PWM Compatibility

- The original controllers were controlled by a potentiometer.
- Solution: L298N H-Bridge



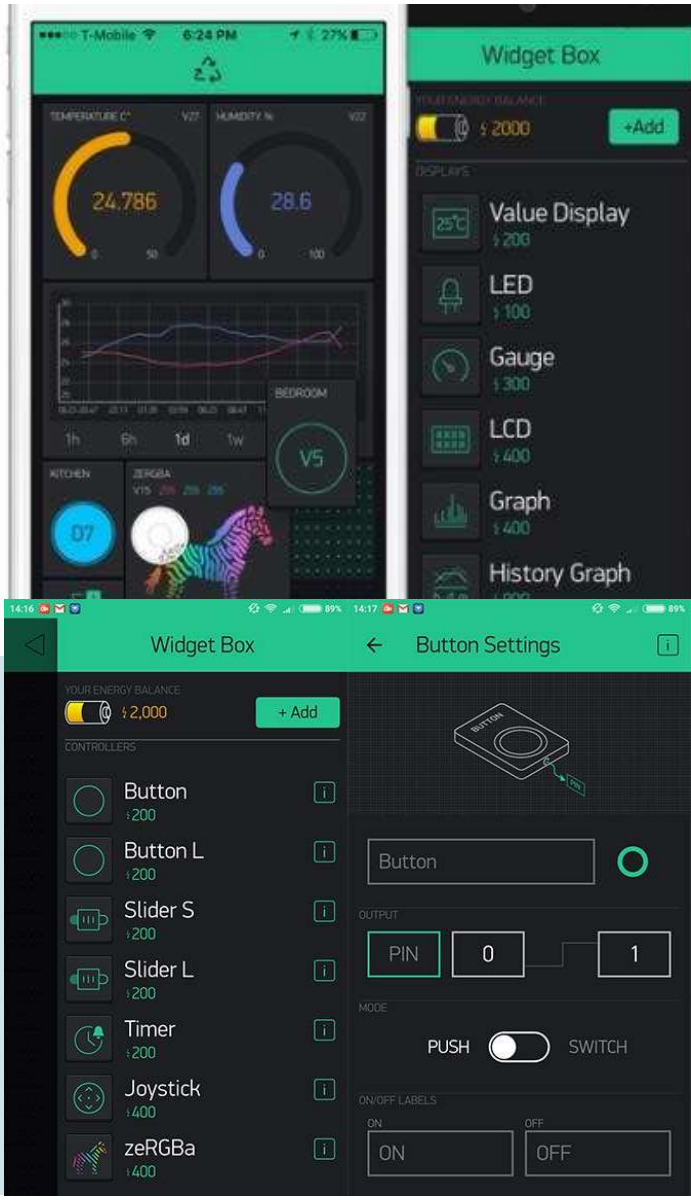
Issue: Vehicle Speed

- Weight, motor type, and voltage supplied made the car too slow.

- Solution: More voltage should be applied to the motors.

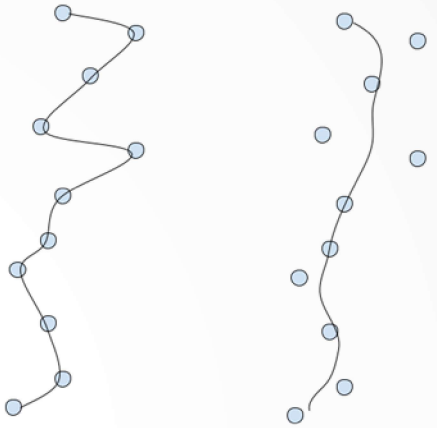
Blynk Mobile Application

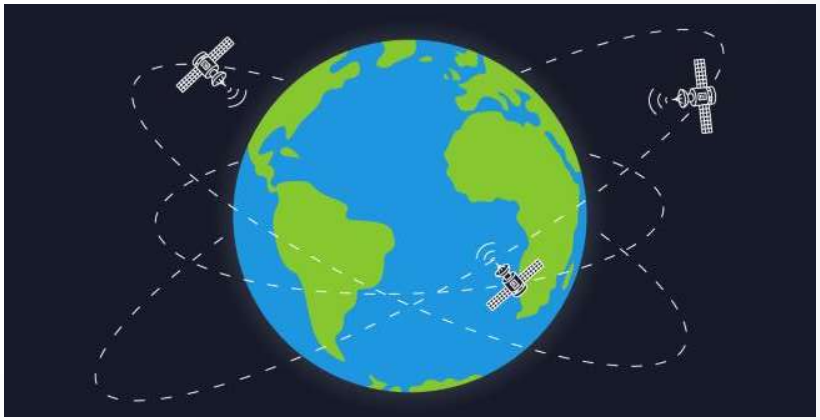
- Idea
  - We planned on making our own app to control SAFER but during our research we found Blynk
  - Blynk allows us to build custom interfaces to control the robot and connects via out Bluetooth module
- Issue
  - The issue comes when the phone was connected for more than 10 seconds
  - After that the connection became unstable. After weeks of trying to fix the issue we decided to write our own app



GPS Following

- Idea
  - The user could use an app to connect to the robot and stream their GPS coordinates via Bluetooth
  - SAFER would have then mimic these coordinates with GPS and Compass modules
- Issue:
  - With the two devices being next to each other the readings where meters apart
  - Even after implementing a rolling averages formula we still had a delta of up to 8 meters.





Administrative Content

Work Distribution-Megan(Update)

	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

Hardware Work Distribution(Update)

	Power Design	Mounting	Wiring	Lighting	PCB Design	Steering	Driving
Megan	2				2	2	
Michael		2					
Jonathan					1		
Jordan	1	1	1	1		1	1

1- Primary
2- Secondary

Software Work Distribution

	Computer Vision	Lighting	Steering Code	Driving Code	Mobile App
Megan		2			
Michael	2		1	2	1
Jonathan	1	1	2	1	2
Jordan					

1- Primary
2- Secondary

Budget

Per Unit Cost

Item	Quantity	Total Cost
Power Wheels	1	$25
Raspberry Pi 3 Model B	1	$35
8 Ohm Loudspeaker	1	$17.95
Rechargeable Searchlight LED Flashlight 6000 Lumens	2	$57.36
16.4 Ft 12 V DC LED Strip Lights	1	$10.68
HC-05 Bluetooth Module	1	$11.11
DC Motors	2	$13.90
12 V Battery	2	$25.90
PCB Estimate	1	$64.06
Perfboard	1	$2.82
10 kOhm Linear Sliding Potentiometer	1	$1.47
PixyCam	1	$59.90
Box for PixyCam Mounting	1	$4.53
H-Bridge	3	$9.60
Relay	1	$2.00
SanDisk Mobile Class4 16GB MicroSD Card	1	$5.31
Miscellaneous(Screws, Nails, etc.)	N/A	$10.62
Total		**$ 200.10**

Total Development Costs

Item	Price/Unit	Quantity	Unit Cost
Power Wheels	~$200-$400	2	$25
Power Wheels Replacement Battery 6V	$35	1	$35
Raspberry Pi 3 Model B	$35	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	2	$22.22
DC Motors	$6.95	5	$ 34.75
DC Motor Controllers	$12.95	5	$ 64.75
12 V Battery from Electric Scooter	$31.79	2	$0
PCB Fabrication	$2	10	$20.51
PCB Components	N/A	N/A	$171.66
Arduino Uno	$18.99	1	$18.99
Perfboard	$2.82	2	$5.63
10 kOhm Linear Sliding Potentiometer	$1.47	2	$2.94
PixyCam	$59.90	1	$59.90
Jumper Wires	$5.79	1	$5.79
H Bridge	$3.20	5	$28.80
Crimping Tool Set	$18.97	1	$18.97
Relays	$2.00	5	$9.99
Box for PixyCam Mounting	$4.53	1	$4.53
SanDisk Mobile Class4 16GB MicroSD Card	$5.31	1	$5.31
Miscellaneous(Screws, Nails, etc.)	N/A	N/A	$21.23
Total			**$ 495.97**

Demonstration Video

In Memoriam

- RIP to all the blown motor controllers and H-bridges
  - 2 DC motors
  - 2 DC motor controllers
  - 3 H-bridges
  - 2 MCUs
  - 1 Sliding Potentiometer

Questions?