# Stiggy

12/6/17

Project Lead: Matthew Conniff
Build Studio Engineers: Daniel Castle, Megan Sapack, Dan Bartels, Chris Bunn, Adam Bennani, Ronit Sharma

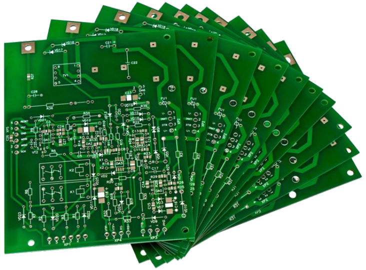
# Stiggy - What is it?

- Developed by Northeastern Professor Stephen Golden
- STEM Learning Toy for young children (K-8th Grade)
- Used to teach fundamental principles of physics and engineering at a young age
- Toy RC car with live sensors and accompanying curriculum

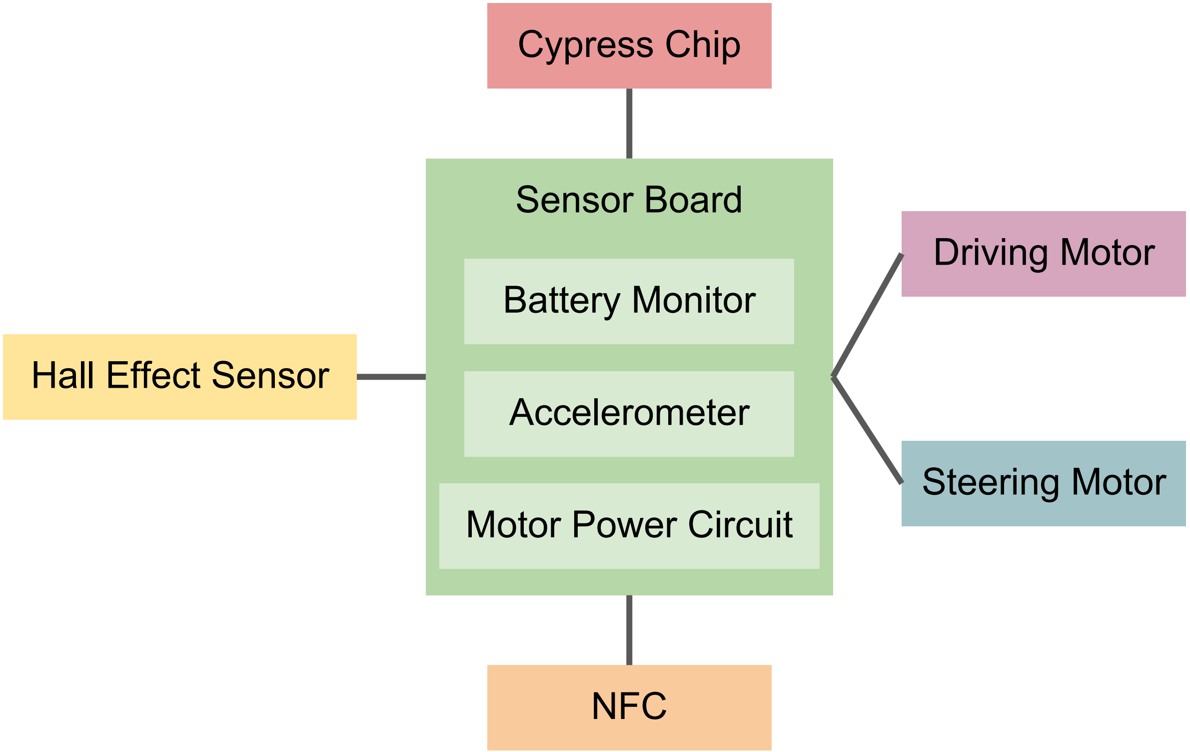
# Scoping

- Started with full car design … in 14 weeks

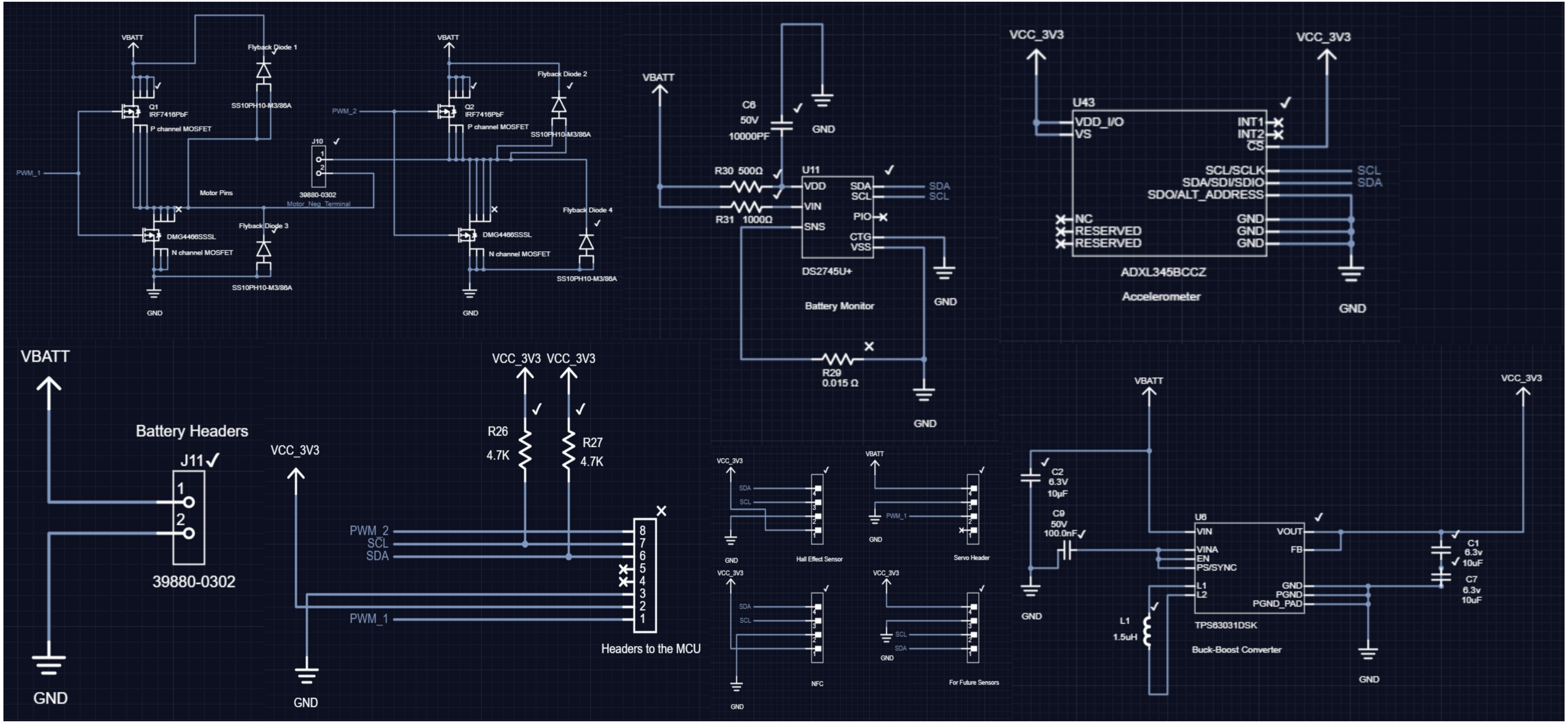
- Decided on “module” with integration mostly a “looks-like”
- Stripped several things out of the scope as we thought they weren’t needed
    - Optical Encoder (swapped for Hall Effect Sensor)
    - Electrical Current Sensor (swapped for Battery Monitor)
    - Gyroscope
    - Barcode Scanner (swapped for NFC)
    - Ultrasonic Sensors



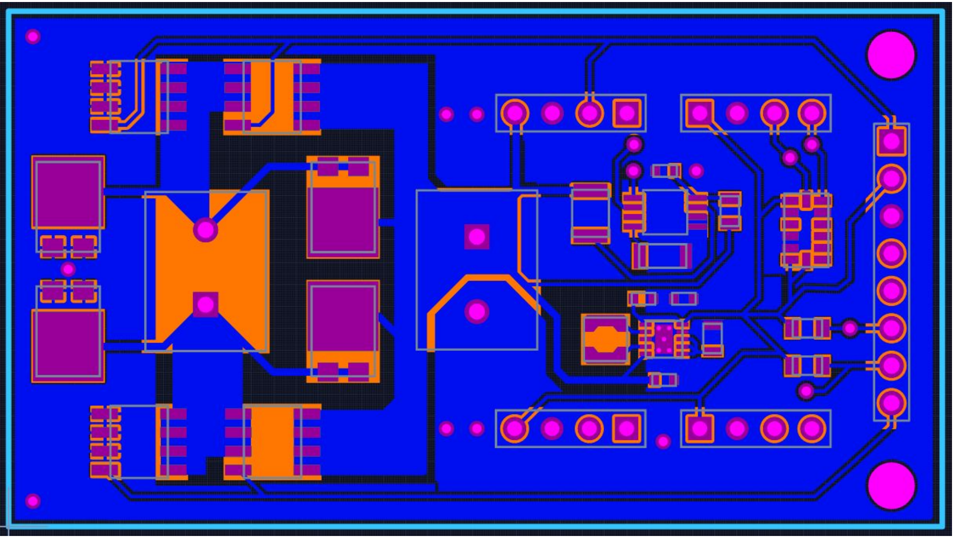
# System Design - Electrical

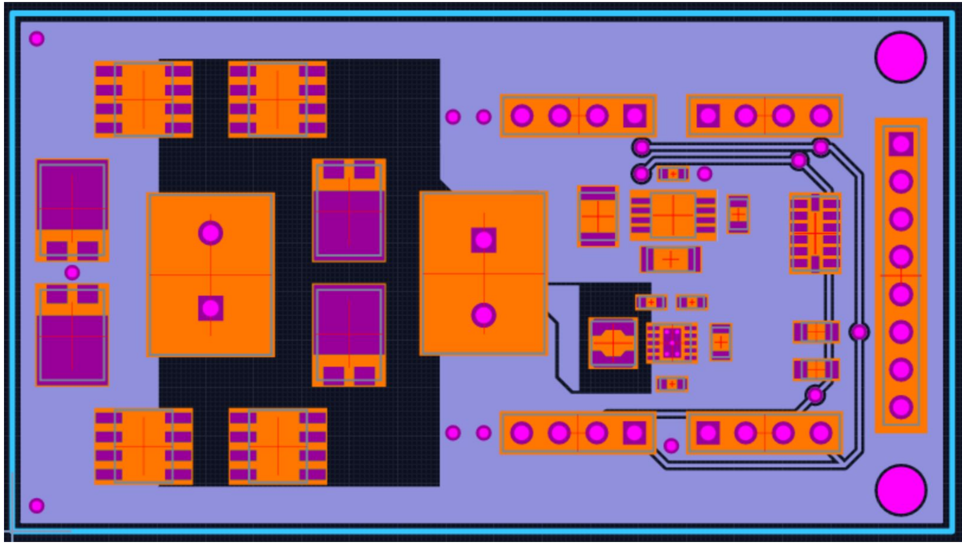


# Electrical Design - Sensor Board

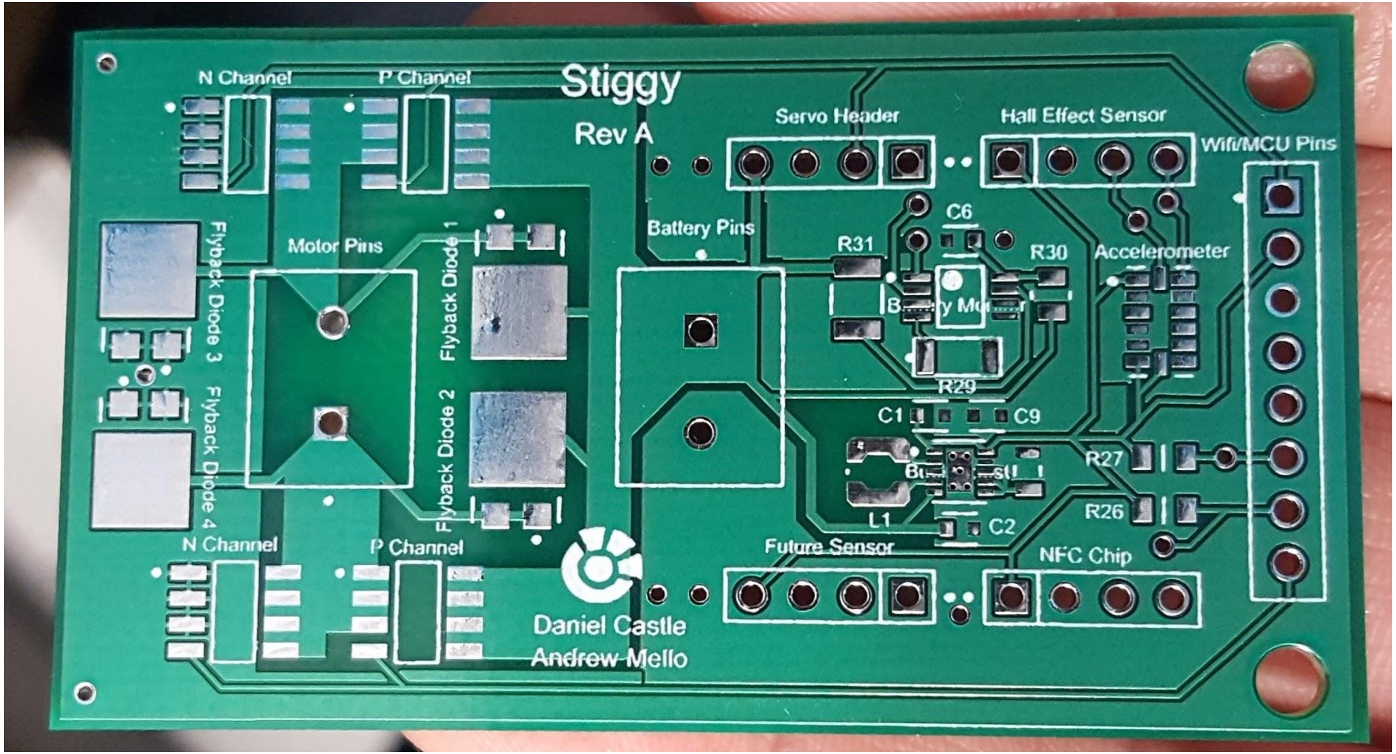


# Electrical Design - Sensor Board





# Electrical Design - Sensor Board

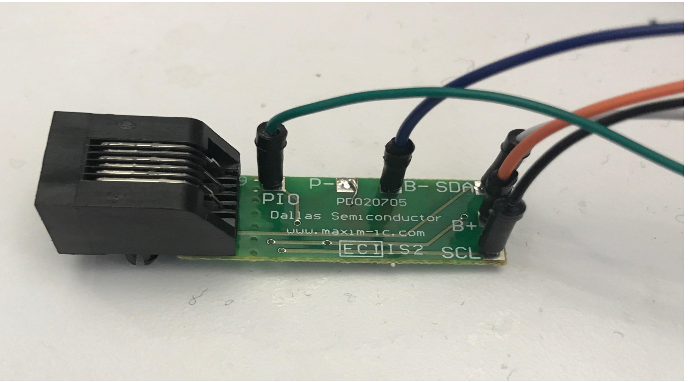


# System Design - Software

- Built utilizing the WICED SDK
  - Compatible with Cypress Evaluation board
  - Allows for setting up network interfaces (such as soft APs) easy
- Focused on integrating multiple different modules and sensors into one device
  - Each module interfaces with evaluation board, which sends as one total package
  - All sensors are able to collect data concurrently and quickly
- Challenges we ran into
  - Porting Arduino compatible libraries to utilize WICED SDK function calls
- Utilizing I2C System Standard
  - Allows for multiple devices to interact over a single serial bus

# Electrical Design - Battery Monitor

- DS2438
  - Maxim Integrated
  - Small size - 3x1 cm
  - Low price - $2.60
  - I2C compatible
- Discharging
  - Battery stays constant, giving off a consistent voltage
  - Towards the end of life, it starts dropping rapidly
- Battery Monitors
  - Connect battery across resistor to act as load
  - Measures voltage across that load
- Power Stage
  - Regulates so that the battery is always giving off 3.3V
  - Reduces power to 3.3V when battery is discharging, and raised it when battery is almost dead



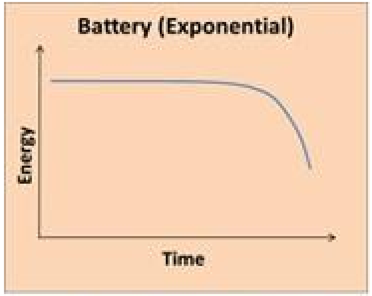
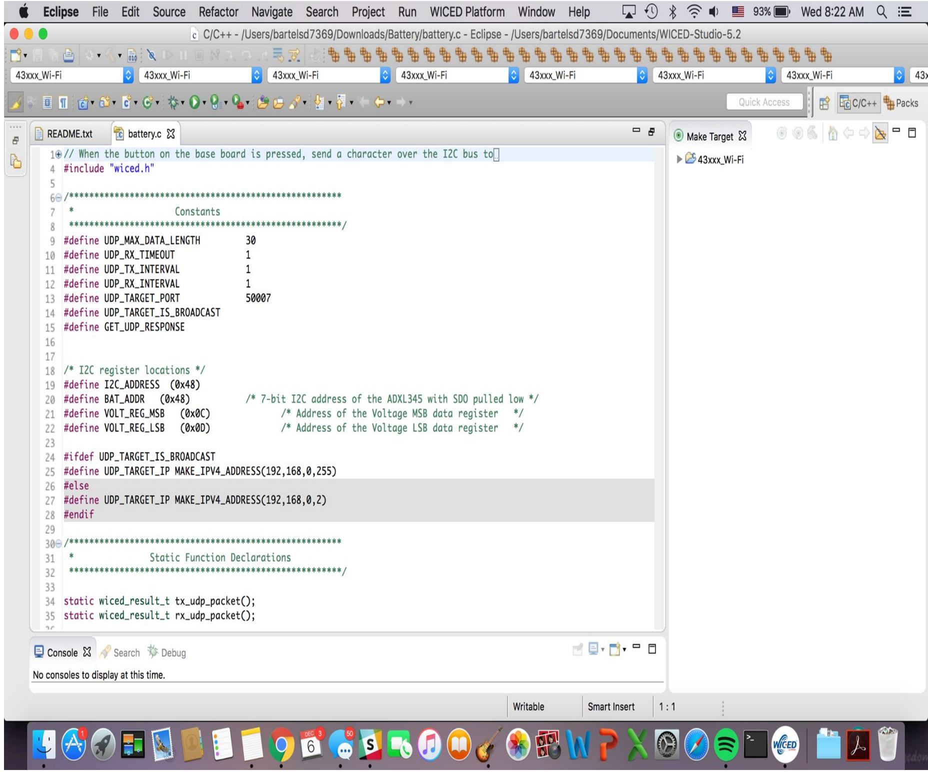




**Figure 1: Discharge curve of Battery.** Exponential discharge provides steady power to the end.

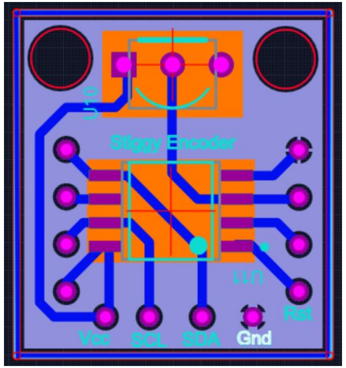
# Software Design - Battery Monitor

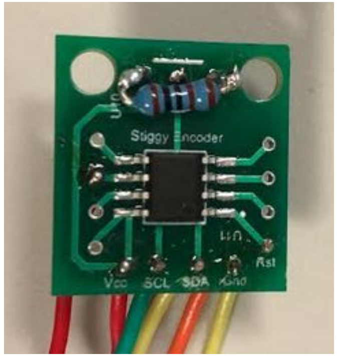
- I2C Registers
  - One register for MSB, one for LSB
  - Battery voltage is stored in the registers
  - Send a write command to choose which register to read from
  - Read command to read that register
  - Shift over a register and read then convert
- I2C Software
  - Establishing master and slave
  - Sending master commands to slave device and reading back results
- Readings
  - Voltages recorded as binary
  - Converted using given table to integer voltages
- Challenges
  - Integration with Cypress Board and UDP
  - Verification of monitor

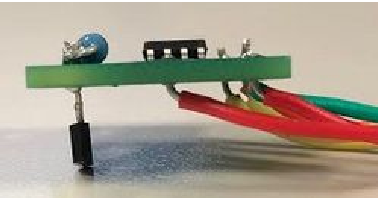


# Electrical Design - Hall Effect Sensor

- Designed a PCB for the Hall Effect Sensor
  - Needed digital output for $I^2C$
  - Microcontroller
  - Latching hall effect sensor
- ATTiny25
  - 2KB ISP Flash memory
  - $0.82
- US1881 Latching Hall Effect Sensor
  - $1
  - Output pin turns high when N pole of magnet passes and low when S pole passes
- Future considerations:
  - Adding a 1K resistor between GND and Vcc of Hall effect sensor
  - Orienting hall effect sensor to be the best for its location

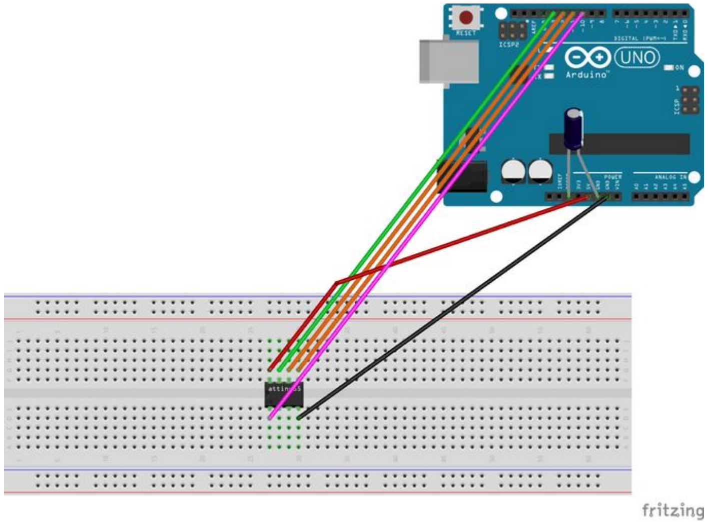


# Software Design - Hall Effect Sensor

- Used arduino as programmer to flash ATTiny
  - Lots of tutorials on how to program
  - Uses arduino IDE
- I²C communication
  - Arduino as Master
  - ATTiny as slave
- Interrupts
- RPM Calculation
  - Can be modified for type of magnet and resolution
- Future considerations:
  - More stable testing fixture
  - Capacity of data over I²C for more than 1 byte



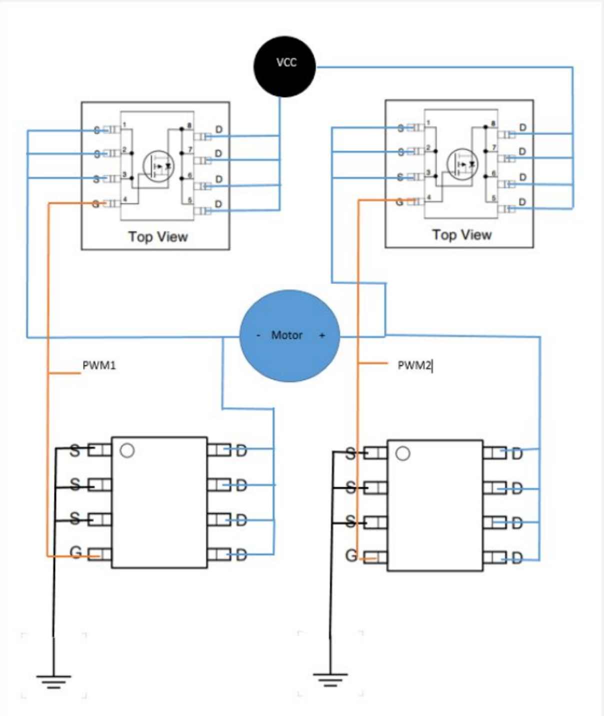
# Electrical Design - Driving Motor

**Problems**

- Initial Stiggy traveled too quickly
  - Tried controlling speed using software
- Car drove out of operating efficiency
  - Sacrificed torque

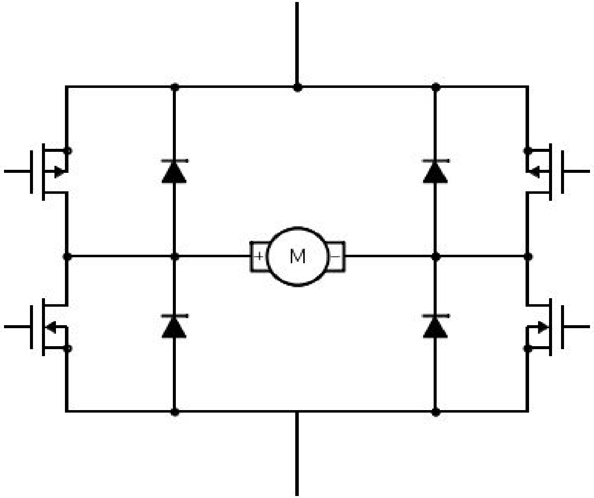
# Electrical Design - Driving Motor

- Researched various drivers for DC Brushed Motors
  - decided to create our own circuit
    - Cheap
    - satisfy high power requirements
- H-bridge configuration
  - one of the most common configurations
  - speed/direction of motor
  - 2 P-Channel and 2 N-Channel MOSFETS
  - 2 PWM signals from Cypress
- Drive motor as efficiently as possible



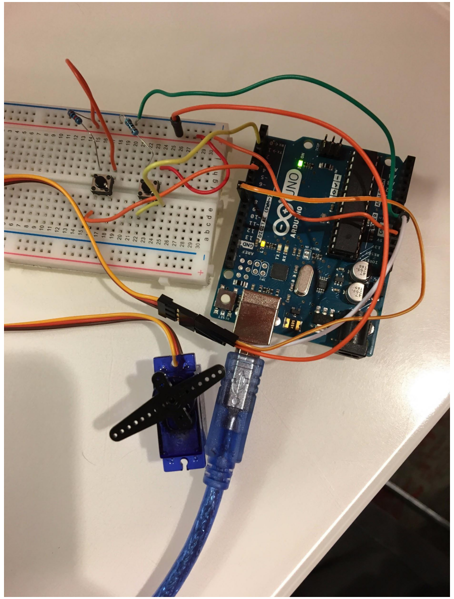
# Electrical Design - Driving Motor

- With the design of the H-bridge spec'd out, there was just one problem:
  - There are sudden voltage spikes across an inductive load when its supply current is suddenly reduced or interrupted
  - AKA, whenever the motor changes direction from forward to reverse, or vise versa
- In order to cope with this "flyback" in our circuit, certain diodes, aptly named flyback diodes, were incorporated into our design
  - These flyback diodes return the inductive spikes to the supply instead of the MOSFETs, which could destroy the transistors if not handled accordingly
- Specifically, Schottky diodes were used, as they have a low forward voltage and an extremely small reverse recovery time, giving a fast switching time at a smaller voltage level



# Software Design - Steering Motor

- 180 positional servo for steering (SG92R Micro Servo)
    - Use a PWM to determine position of servo
    - Operate by sending timed +5V pulse
    - Corresponds to servo position
        - 5V for 500 microseconds = 0 degrees
        - 5V for 1500 microseconds = 90 degrees
        - 5V for 2500 microseconds = 180 degrees
        - Relationship is linear
- Control interface using two push buttons
- Similar to a slider on a phone



# Software Design - Wifi Controls

- Creation of soft AP to connect with host device
- Transmission of driving signals and car information using UDP packets
  - Reduces latency compared to TCP protocol
- Raw data values are sent directly to host to process
  - Acceleration, battery level, etc.
  - Processing / calculations are done on the host device
- Packet is formatted to deliver data in 50 ms intervals and to process driving packets as soon as possible
  - Data transmission: `Stiggy: x: 0.5 y: 0.5 z: 0.5 b: 2.9`
  - Driving packet: `Stiggy: L`

# Software Design - Accelerometer

- Tracks the acceleration of Stiggy (in Gs)
  - Can calculate velocity and displacement
- Sparkfun ADXL345 chosen
  - Documentation and community support
  - Expansive feature set
- Communicates utilizing I2C protocol
- Possible expansion of features into future:
  - Using single/double tap detections to detect collisions

# Mechanical Design

## Motor Related Items

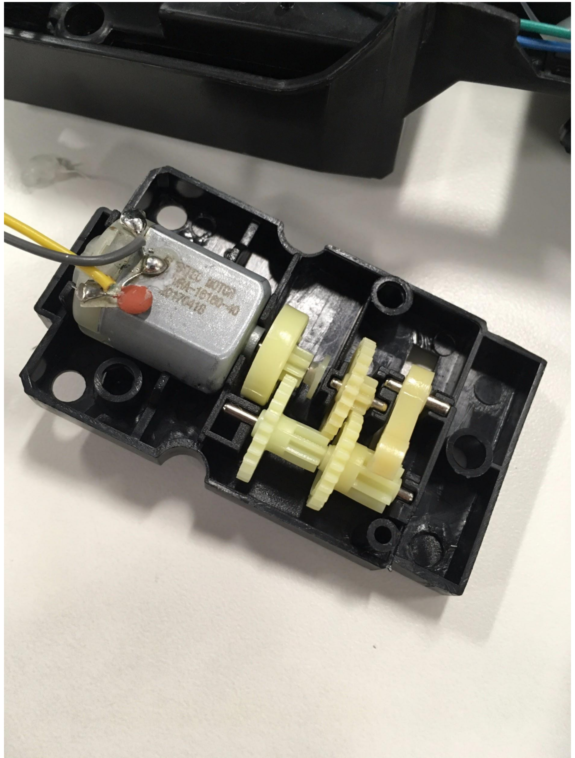
- Stiggy Requirements
  - Speed of 8-10 ft/s
  - Able to climb a 30 degree incline
  - 5cm Wheel Diameter
- Conducted experiments to verify the motor and battery were compatible
- Ran calculations to provide client with necessary information to select a motor from their manufacturer
  - 3V DC Brushed Motor
  - RPM range ~ 932.04 - 1164.25 rpm
  - Minimum Torque ~ 162.24 mNm

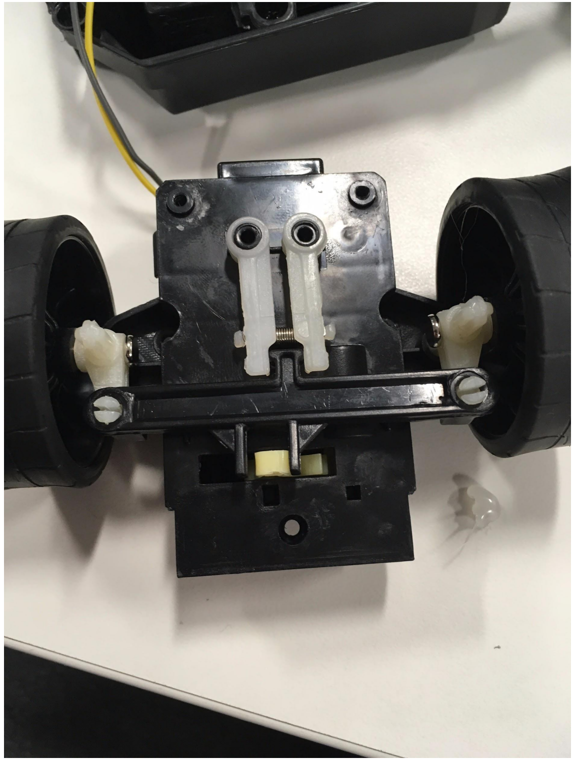
# Mechanical Design cont.

**Integration**

- Conducted an initial assessment of the car including similar power system analysis
  - Ran experiments to verify the car would run within Stiggy's desired specifications and would therefore be a good representation
- Retrofitted the steering system to be powered by a servo motor
  - For prototyping purposes using

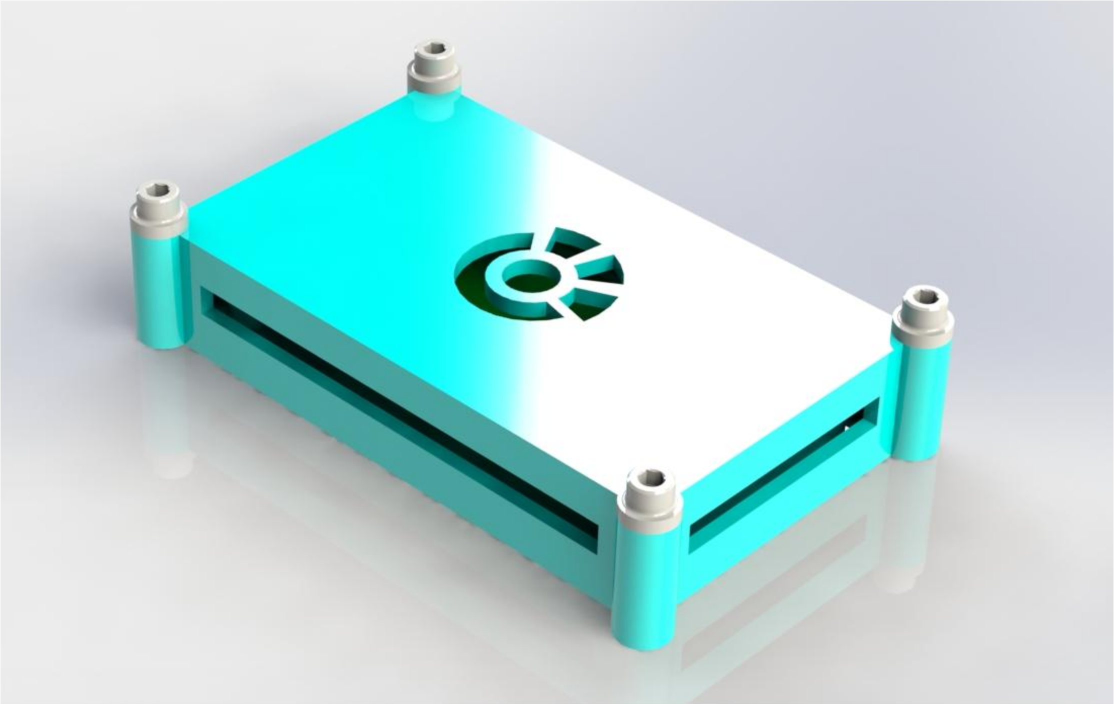
# Mechanical Design cont.





# Mechanical Design cont.

- Designed a case for the car’s PCB



# Mechanical Design cont.

- Mechanical System Layout
    - PCB mounted on top of car for prototyping purposes, but can be integrated into the chassis
    - NFC needs to be on the underside of the car preferably near the front axle
    - Hall Effect Sensor mounted near one of the two rear wheels

# Next Steps

- Testing/Iterating of Sensor Board
- Integration with real RC car
- Sourcing chassis from China
- Industrial Design