Autonomous Line Following Racecar

EE192 Spring 2018, Professor Ron Fearing

Trey Fortmuller, Joey Kreoger, Charlene Shong (Team 6)

February 3, 2018

1 Overall Strategy

1.1 Winning

We seek to deliver a reliable, well tested autonomous vehicle by the end of the semester. In order to make this project's development process as painless as possible, we plan on only using hardware supported by the EE192 staff. This practice keeps cost down and ensures we have an experienced GSI to consult throughout the semester. In the spirit of "measuring twice, and drilling once," we'll spend extra time on our control system verification in simulation, and develop custom real time telemetry visualization tools to run our laptops. This will allow for quicker debugging and a greater understanding of our vehicle's processes during testing.

Essential Features listed below includes upgrades we plan on making to the basic vehicle construction that do not add significant complexity, but offer important performance or usability improvements. *Possible Upgrades* listed below includes upgrades that may offer significant performance improvements, but introduce a complexity to the system that may become unmanageable. We plan on only approaching these improvements if we find ourselves ahead of schedule throughout the development process.

Essential Features

- RTOS pseudo-threading management.
- Wireless emergency stop switch.
- Line camera for track detection and steering control.
- Motor encoder for speed detection and throttle control.
- Single transistor motor drive circuit.
- 3S LiPo battery with voltage telemetry and auditory low voltage warning as prescribed in the NatCar rules.

Possible Upgrades

- Implement an optimal control strategy via LQR.
- Two-transistor motor drive circuit to allow braking.
- A long range and short range line camera for better trajectory planning.
- All-wheel drive configuration for improved cornering performance.
- Real time vehicle telemetry via bluetooth and graphical ground station tools for monitoring and debugging.

- Upgraded embedded computer to quad-core ARM64 CPU (see the Nvidia Jetson TX2) allowing for true multithreading, and real time video processing with a typical webcam rather than the line camera for superior state estimation.
- Implement iterative learning control to improve performance around a given track with each lap.
- Brushless motor and electronic speed controller for efficiency and decreased electronic noise over brushed motors.
- Include an optical flow sensor for speed estimation independent of motor RPMs

1.2 Racing

Our racing strategy will be to prioritize accuracy over speed. A slow time is a far more preferable result than losing localization about the track or crashing out. We plan on implementing optimal control via LQR and analyzing the controller's performance in MATLAB simulation prior to test drives. An optimal controller will be one that maximizes average lap speed whilst navigating the track with precision. We'll quantitatively analyze our vehicle's performance as we tune our controller based on lap times, track divergence, and acceleration profiles. We plan on synthesizing weighted control input based on trajectories generated from the long range and short range cameras to follow curves smoothly. A principal goal of the team will be to develop an easily portable and modular codebase such that as many racing behaviors as possible (braking timing, corner cutting, etc.) are easily modifiable.

1.3 Sensors

- Long range camera
  - A 128px line camera used to determine the direction of the track far ahead of the vehicle with the respect to the vehicle frame. How *far* ahead of the vehicle the camera detects the track is a function of the tilt angle of the camera, an adjustable parameter we will tune during testing. This data will be an input to our steering control.
  - Requires one analog input to the K64 microcontroller.
- Short range camera
  - A 128px line camera used to determine the direction of the track immediately ahead of the vehicle with the respect to the vehicle frame. How close to the car the camera detects the track is a function of the tilt angle of the camera, an adjustable parameter we will tune during testing. This data will be an input to our steering control.
  - Requires one analog input to the K64 microcontroller.
- Motor encoder
  - A motor encoder attached to the brushed motor which propels our car, likely an incremental optical encoder. This gives us data regarding the rotation frequency of the motor which can be transformed to vehicle speed, an input to our speed controller.
  - Requires one digital input to the K64 microcontroller.

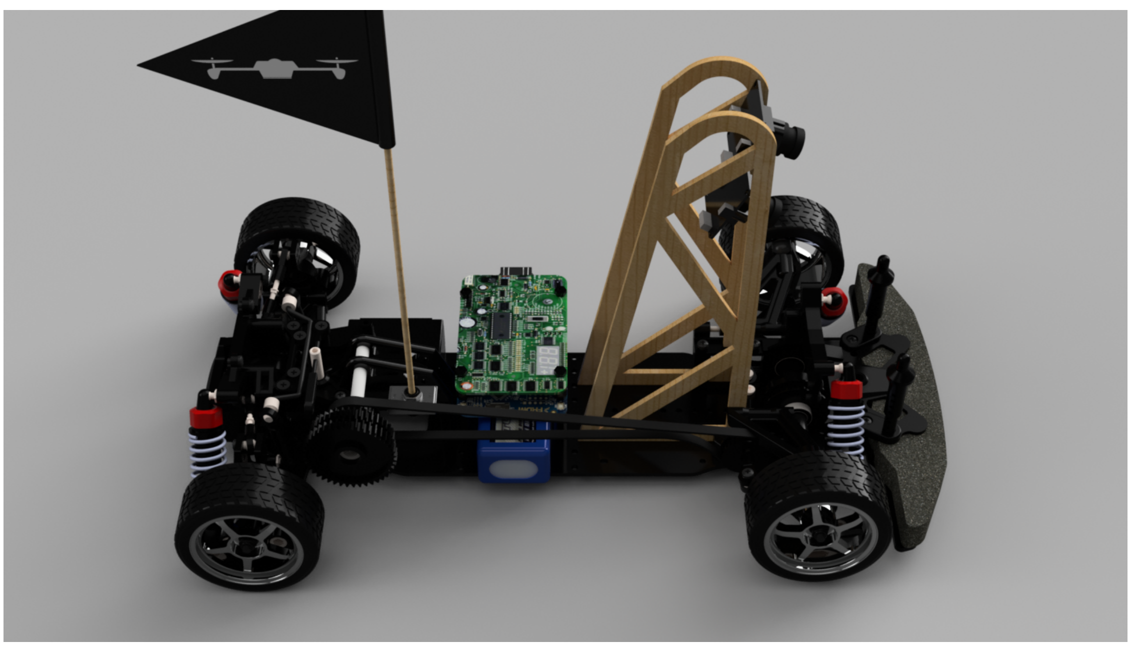
2 Hardware Design

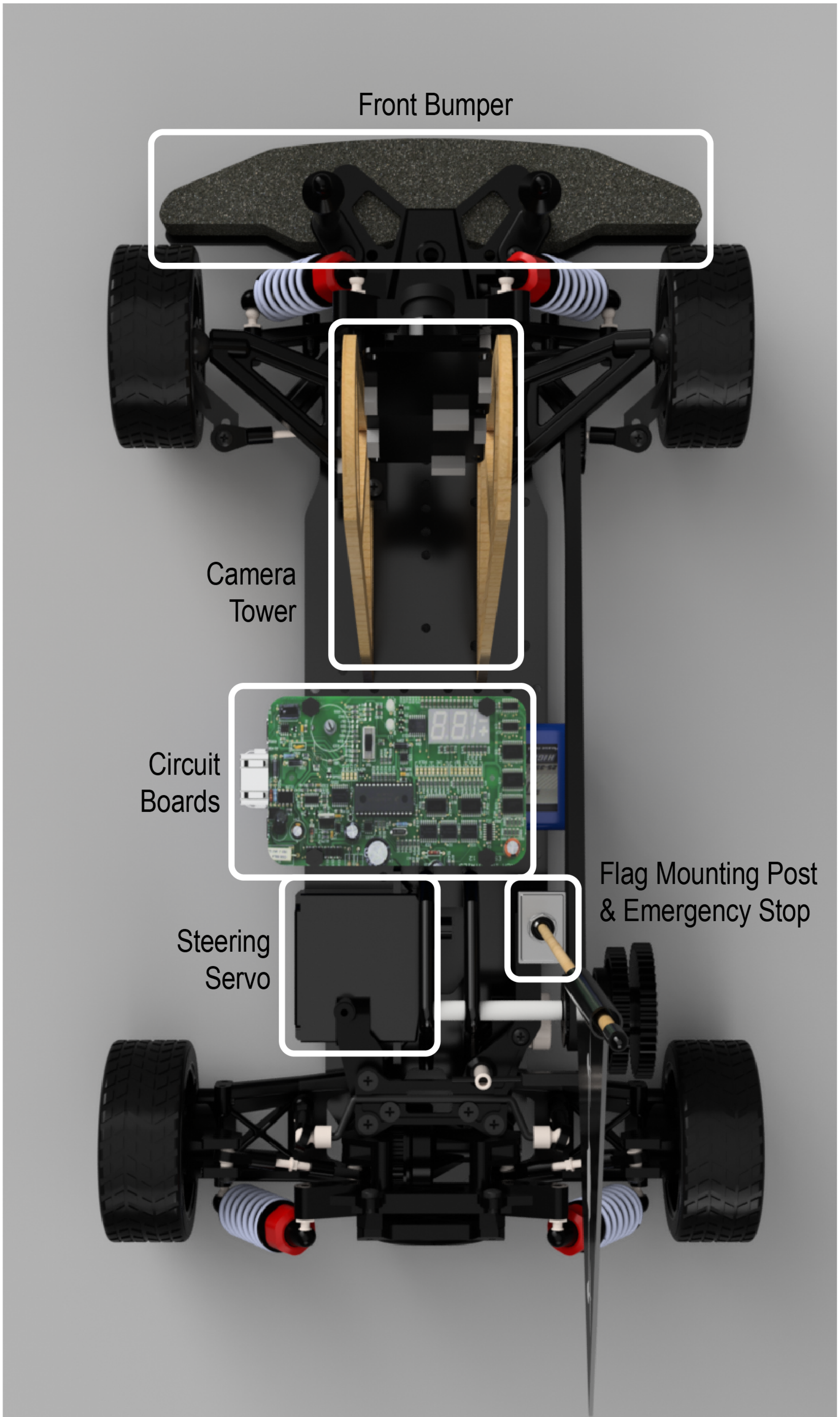
2.1 Attachments to Vehicle

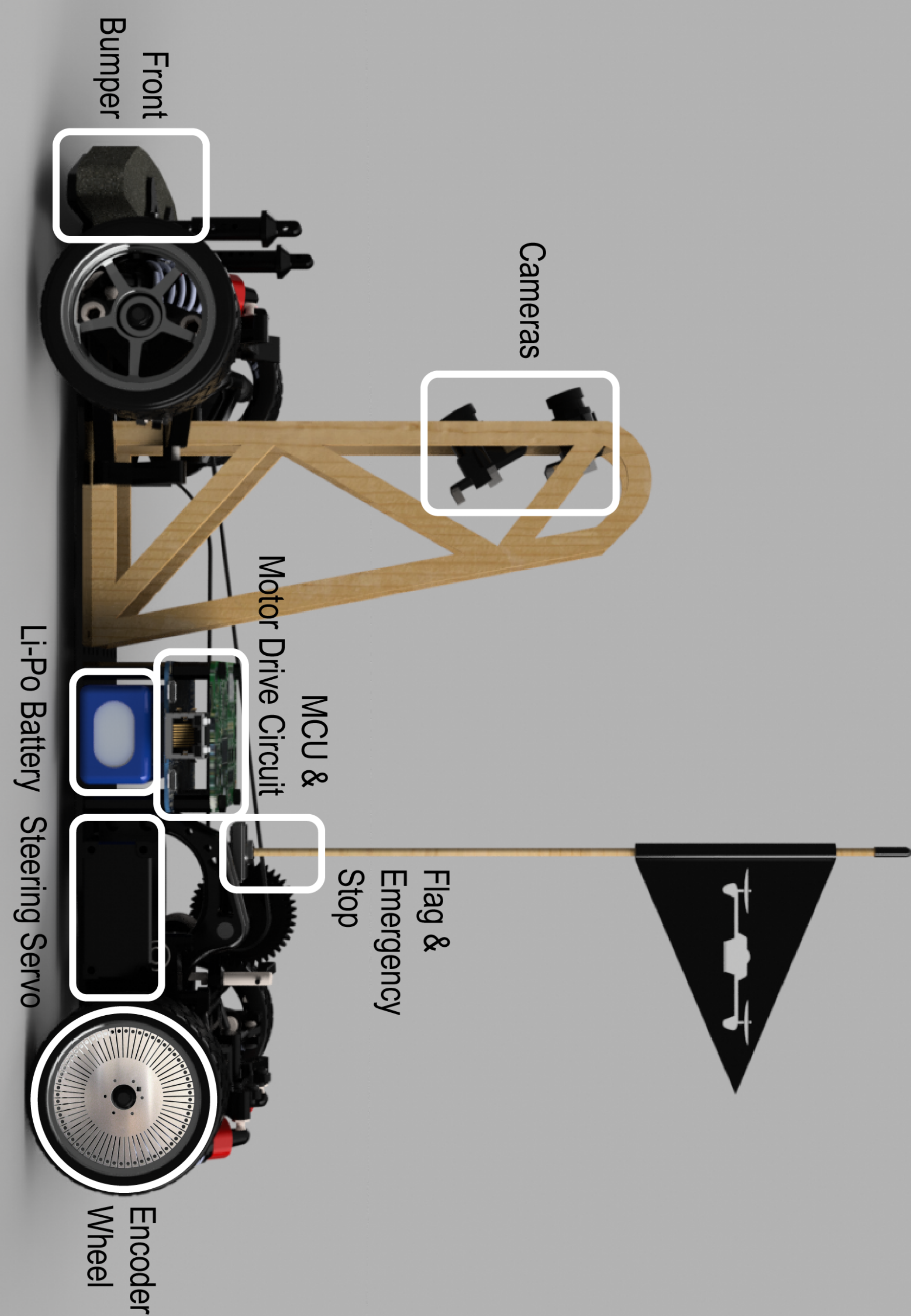
- FRDM-K64F microcontroller
- Motor drive circuit board
- DC brushed motor
- Motor encoder
- (2) Line cameras on a mast
- Line camera adjustable tilt mounts
- Bluetooth transceiver module
- flag and mast
- E-stop switch
- Rest switch
- 3S Lipo battery
- Steering servo and connections
- Timing belt and gears for all-wheel-drive
- Beeper for low voltage alarm

2.2 Mechanical Drawings of Vehicle

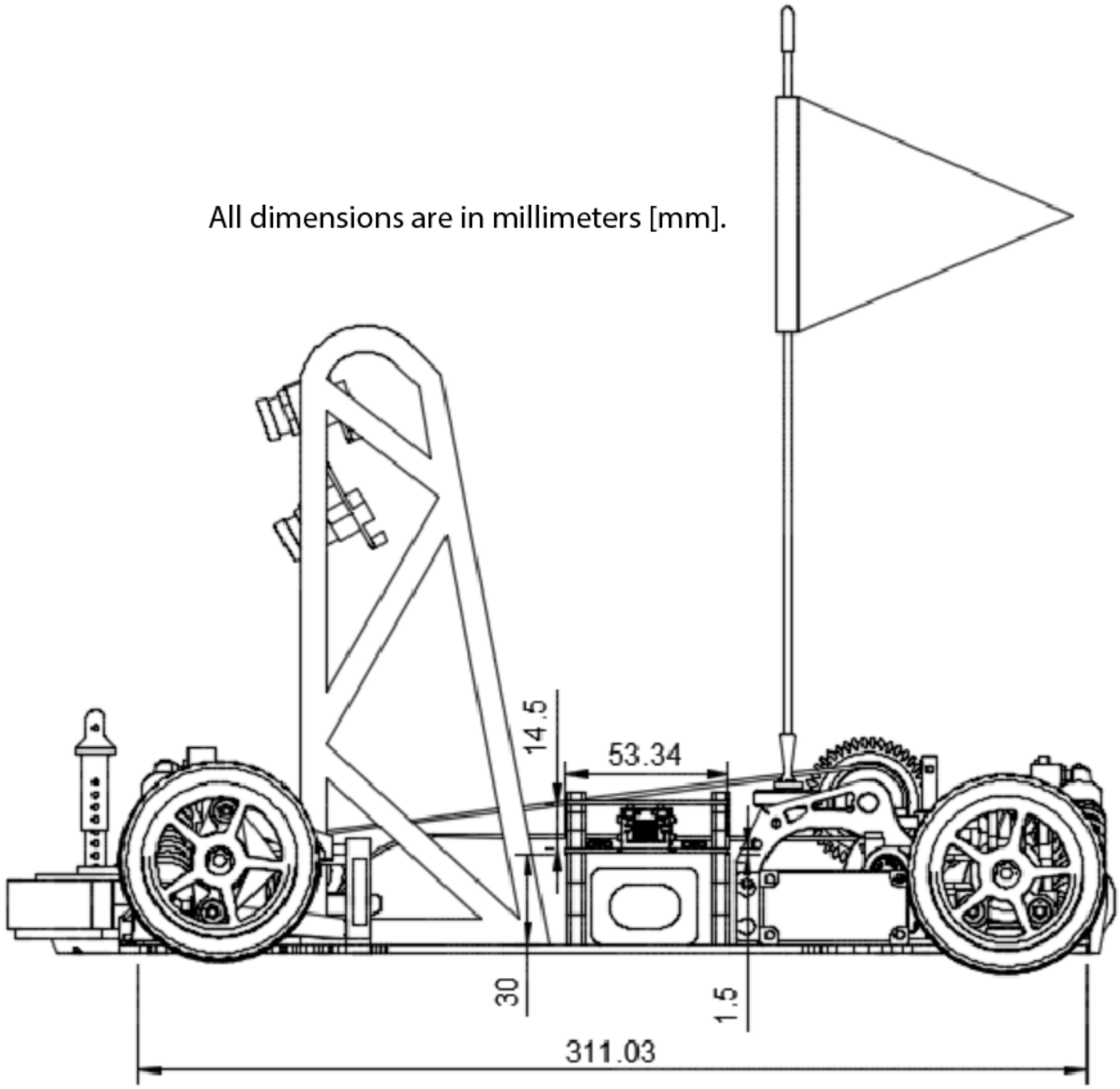
We decided to jump straight to 3D CAD for the vehicle to save time in future weeks. The current design files are based off the 2D drawings of components in the manual for the new chassis as well as existing 3D CAD data found online for previous models of the car. We rendered the files in Autodesk Fusion 360.




Front Bumper
Camera
Tower
Circuit
Boards
Flag Mounting Post
& Emergency Stop
Steering
Servo


Front
Bumper
Cameras
MCU &
Motor Drive Circuit
Li-Po Battery
Steering Servo
Flag &
Emergency
Stop
Encoder
Wheel

All dimensions are in millimeters [mm].



2.3 Special Materials

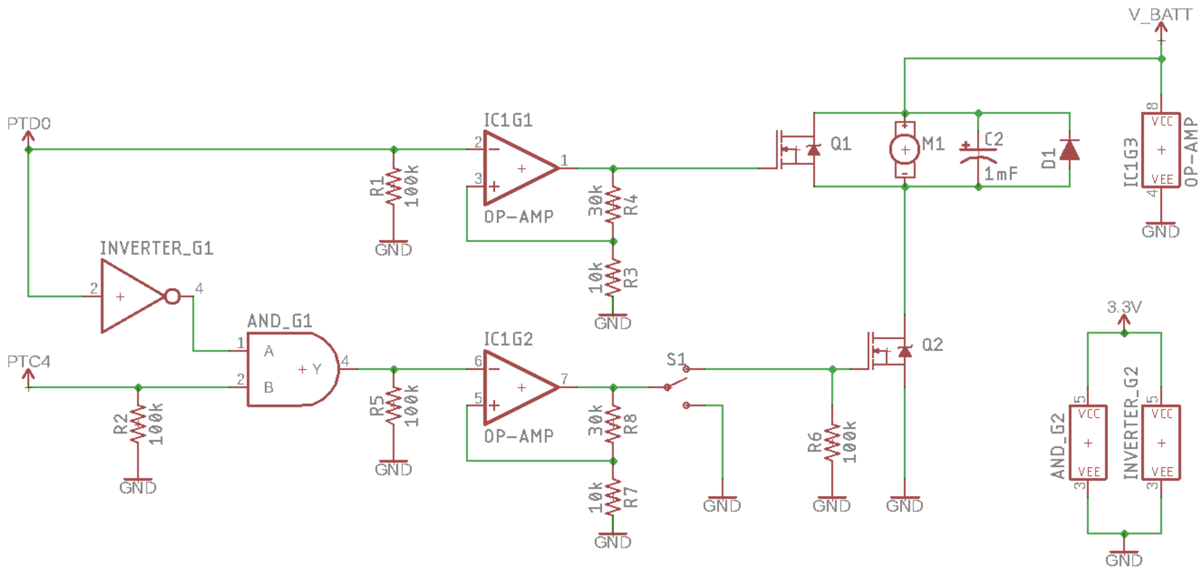
As the car is built up and our mechanical design is developed beyond the basic rendering above, we may include custom parts for component mounting and vehicle structure. Our team has access to a 3D printer capable of printing in thermoplastic polyurethane, nylon, carbon-reinforced nylon, and other high strength materials should we need to fabricate custom parts with complex 3D geometry. Through Jacobs Hall, our team can laser cut 1/8th inch plywood or acrylic, and water jet 1/8th inch aluminum plates. These powerful CNC tools offer rapid prototyping capabilities allowing us total freedom over the mechanical layout of the car.

We plan on securing components on our car with standard M3 fasteners and nylocks. M3 aluminum stand offs and nylon stand offs will be used throughout the vehicle to secure PCBs to the mechanical structure of the vehicle. All these fasteners and stand offs will be purchased through McMaster-Carr and Amazon. Extra switches, connectors, and miscellaneous electronic components will be purchased via Digi-Key and Amazon when not readily available on campus. Our team will take advantage of class purchases to Digi-Key when reduce cost or lead time.

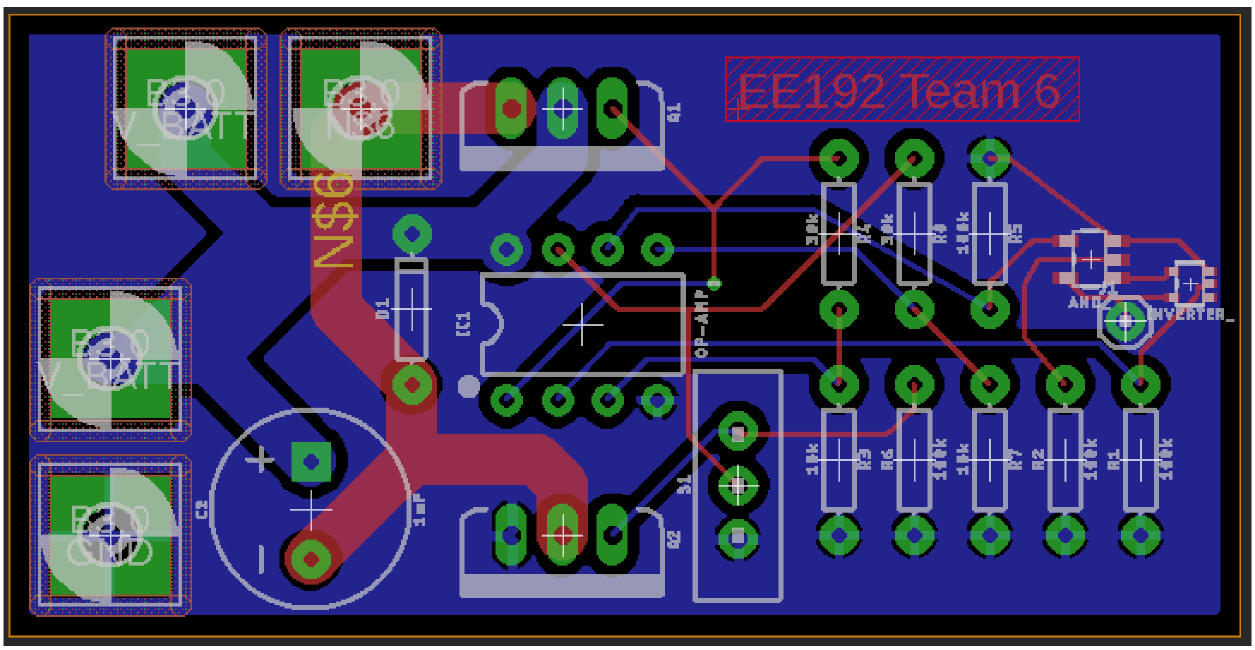
- 1/8th inch plywood camera tower
- 35mm aluminum stand offs
- Nylon stand offs (assorted height)
- 3D printed component mounting parts
- Velcro straps for securing battery
- Adhesive foam for battery protection
- Assorted M3 fasteners (6mm to 30mm)
- M3 nylocks
- Wooden dowel for flag stand
- Construction paper for flag
- Heatshrink throughout
- Double-sided foam tape

2.4 Motor Drive Circuitry

2.4.1 Schematic

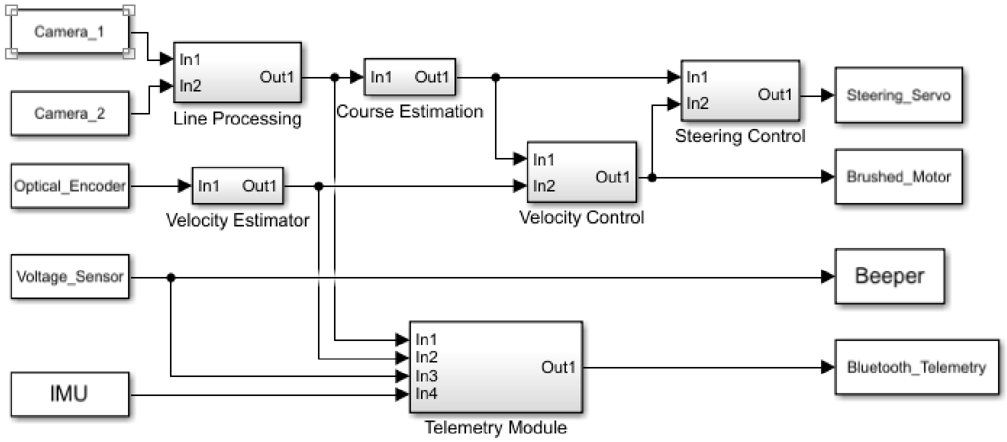


2.4.2 Layout



3 Software Strategy

3.1 Block Diagram



3.2 Module IO

Module	Input	Output	Execution	Execution Rate
Velocity Estimator	Optical Encoder	Velocity Control, Telemetry Module	RTOS Timer	200 Hz
Velocity Control	Velocity Estimator, Course Estimator	Steering Control, Brushed Motor	Interrupt	40 Hz
Line Processing	Camera 1 & 2	Course Estimation, Telemetry Module	RTOS Timer	200 Hz
Course Estimation	Line Processing	Velocity Control, Steering Control	Interrupt	40 Hz
Steering Control	Course Estimation, Velocity Control	Steering Servo	Interrupt	40 Hz
Telemetry Module	Line Processing, Velocity Estimator, Voltage Sensor, IMU	Bluetooth Telemetry	Thread	N/A

3.3 Timing Diagram

<table>
<tr><td>1</td><td></td><td></td><td></td><td></td><td colspan="5">Data processing/estimation</td><td colspan="5">PID controller</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>2</td><td colspan="4">Image capture</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="3">Actuation</td><td></td><td></td><td></td></tr>
<tr><td>3</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="3">Telemetry interface</td></tr>
<tr><td></td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td></tr>
</table>