ECE445 FALL 2025

SENIOR DESIGN LABORATORY

PROJECT PROPOSAL

Omni-directional Aerial Vehicle

Team:

Ivan Ren - iren2@illinois.edu

Dhruv Satish - dsatish2@illinois.edu

Mahir Koseli - mkoseli2@illinois.edu

TA:

Jason Zhang - zekaiz2@illinois.edu

Contents

1. Introduction

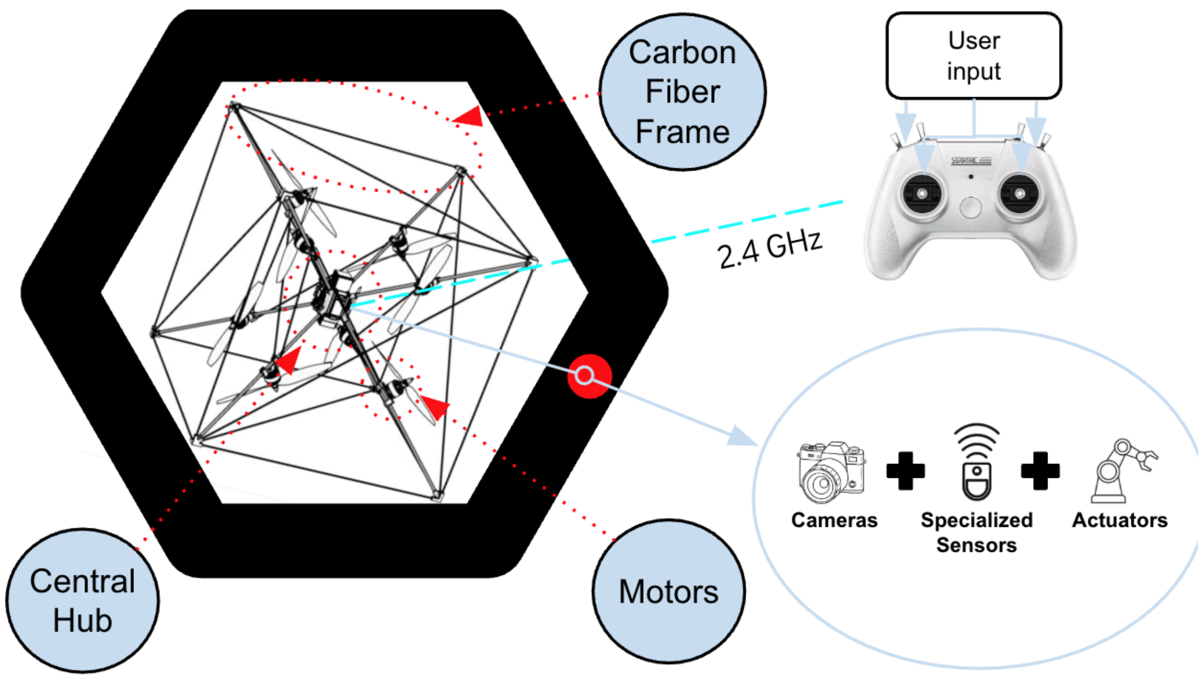
1. 1 Problem

The issue of aerial maneuvering has become an increasingly important consideration in the new age of drone deliveries, drone imaging, and necessity for automation in the fields of agriculture, construction, surveying, remote monitoring, and more. The current standard of drone technology remains limited to mostly quadcopters, a technology that has matured to enough of a degree to allow for complex directional motion, and extreme speed and stability. However, these vehicles have a notable issue of a lack of movement decoupling, with the translational and rotational motions being tied together. In a lot of speed-focused applications, this issue is trivial as most movement systems can compensate to move in 6DOF space by applying different amounts of power to different motor configurations. But in precision applications or in situations that require a certain orientation to be held, decoupling the rotational and translational degrees of motion allow for the drone to have unprecedented control. For example, in an omnicopter design by ETH Zurich, their demo of catching balls using a net showed impressive results, with the drone staying motionless midair while rotating to track the ball and catch it with a net (reference). Just considering a few simple scenarios, for precise filming, construction, or especially sensitive natural or urban areas, a drone with full control over its movement means the ability to hold an angle for a shot, to apply paints at all angles and move around objects through very tight spaces, or to survey wildlife or urban areas without interfering with the natural environments. In any situation not prioritizing speed or power, an omnicopter would provide significantly improved flexibility and control.

1.2 Solution

Our solution consists of three main components: build a robust motor drive system from scratch along with a regenerative braking solution in an omnicopter configuration, designing and 3D printing a frame with the required orientation of motors, and creating the required controls and communications to move the drone in both translational and rotational directions. The motor drive system will contain all required electronics to power and control the motors, including the ESCs, motors, current and voltage sensors, battery management system, and a central microcontroller that interfaces with the ESCs and remote controller. The system will be built to be modular, with each ESC and motor addition being its own module and being easily added to the overall electrical schematic to ensure flexibility with motor configuration, depending on power usage during testing. Within the motor drive system, the battery management system and regenerative braking feature will store away extra power produced by the large currents and wattages that spike up from the motor's inductive nature. The frame of the omnicopter will take the form of either a 6 or 8 motor configuration depending on power draw, stability, and feasibility testing after the electronics have been developed. The design will place an emphasis on easy fabrication using quick prototyping methods like FDM 3D printers, while also remaining lightweight and structurally sound. The goal here is for the drone to be easily manufacturable by hobbyists who would like a robust omni-directional drone with all required functionality and maximum tinkerability. The communications and controls side will handle reading and writing data from the drone to the remote controller, as well as converting decoupled movement signals into different motor power combinations to enable separate translational and rotational movement. The remote controller will be a simple dual-joystick system with each joystick handling either rotational and translational motion. This system also includes the inertial management unit required to track current orientation and balance the system once controls have been input, as well as the antennas required to communicate with the remote controller. Depending on time constraints, trajectory planning and more can also be explored with this side of the project by using the drone's initial position, motor velocities, and orientation. The final solution will consist of a multi-rotor drone capable of separate rotational and translational flight powered through onboard battery packs, responding to inputs from a remote controller through 2 joysticks controlling rotation and translation independently.

1.3 Visual Aid



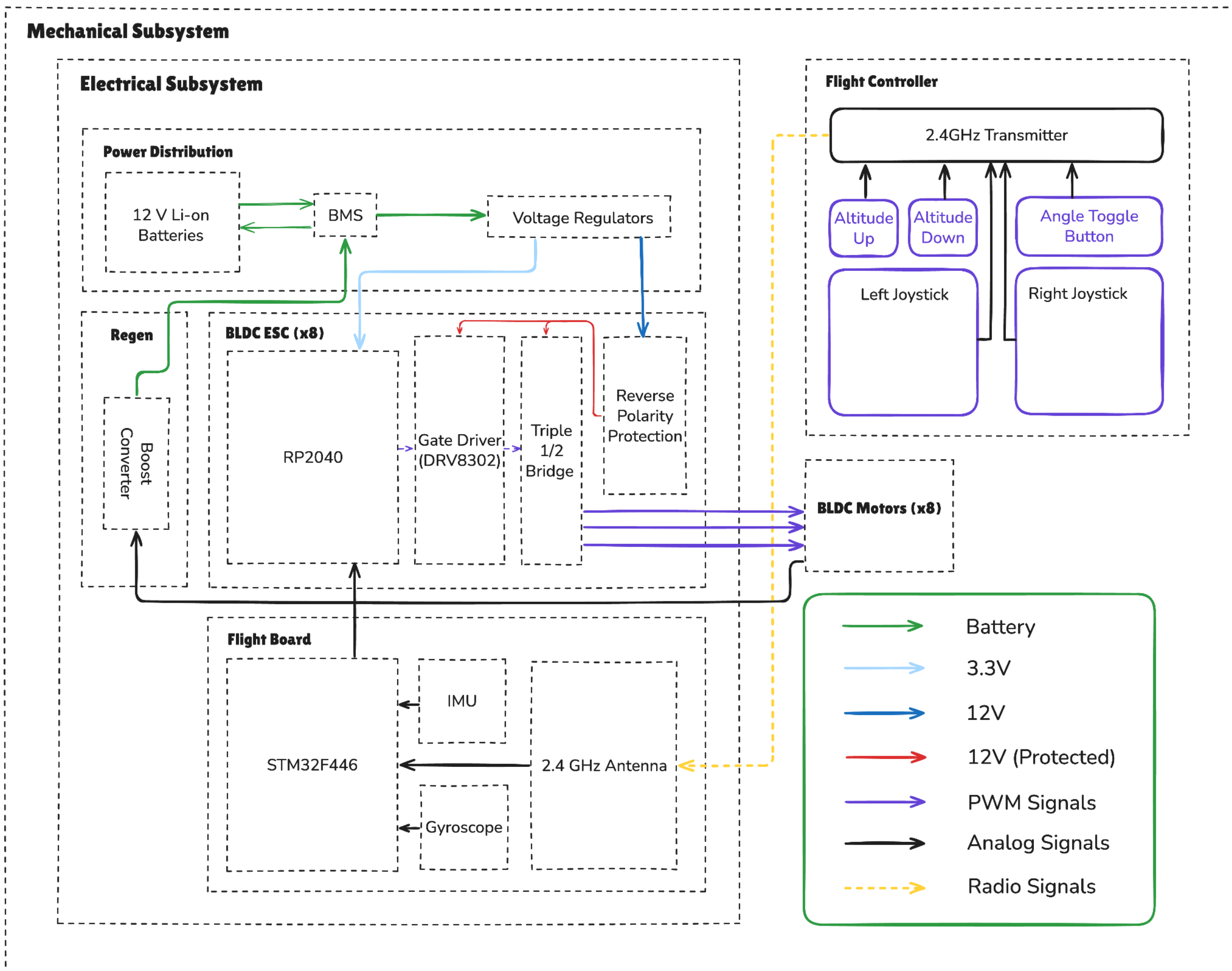
1.4 High-level Requirements

- ☐ Individual ESC able to ramp up and down BLDC motor through PWM control
- ☐ Physical design with optimal motor orientation and mounted electronics/motors
- ☐ Move (1) vertically upwards **1 meter**, hold (2) **a fixed altitude** ± **0.25 meter**

Between each of these goals, we'll have checkpoints to unit test the PCBs, power systems, antenna communications, mechanical frame, and more. For example, confirming each ESC can receive PWM signals from a function generator or Arduino and then characterizing the output motor speed in respect to current and voltage draw will be a single step in our testing process. This organically leads to flight control next, ensuring our microcontroller can receive commands remotely.

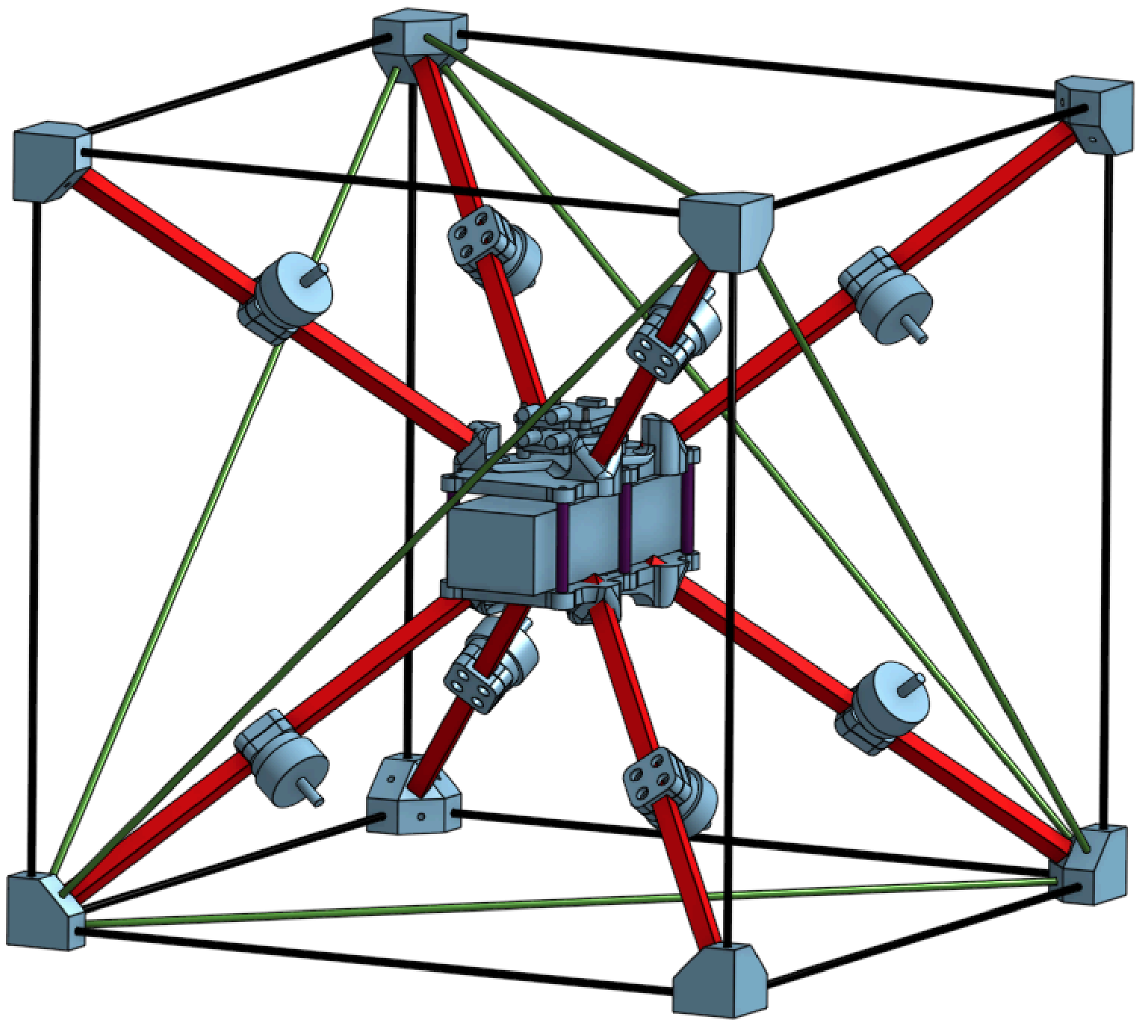
2. Design

2.1 Block Design



Our overall system can be broken into 3 main parts: the mechanical subsystem, electrical subsystem, and flight control/telemetry subsystem. This block diagram illustrates the overall connections between each subsystem. In particular: the mechanical subsystem forms the frame for the electrical subsystem to attach to, the electrical subsystem handles all motor control and power management, and the flight control/telemetry subsystem reads user input and translates said input into an executable output through a hardware interface.

2.2 Physical Design



This design is taken straight from an example project based off of the ETH Zurich paper, in our case, some adjustments will be made to accommodate different carbon fiber rod dimensions, and a different central controller. The mechanical design of the drone can be roughly divided into a few sections: a central hub where the battery/sensors/electronics are stored, 8 arms to hold each motor in a specific orientation, and carbon fiber rods/connectors to create an external frame for structural rigidity/base to land on. More below in the Subsystem Overview.

2.3 Subsystem Overview/Requirements

2.31 Electrical Subsystem

The electrical subsystem will contain all required electronics to power and control the motors, including the ESCs, motors, current and voltage sensors, battery management system, and a central microcontroller that interfaces with the ESCs and remote controller. The system will be built to be modular, with each ESC and motor addition being its own module and being easily added to the overall electrical schematic to ensure flexibility with motor configuration, depending on power usage determined during testing. In specific, each ESC has:

- 6 MOSFETs in a triple half-bridge configuration connected to a main gate driver (like TI's DRV8302) to provide the required control per phase for each 3-phase BLDC motor depending on a PWM signal from the MCU.
- A sensor-less back EMF (BEMF) and phase voltage measurement system needs to be in place to approximately track the position of the rotor in each motor to know when and how much current needs to be applied to each half-bridge. This requires a voltage-divider network setup up in a wye-formation, mirroring the 3 phase setup of a BLDC motor and allowing us to measure both the individual phase voltages and also creating a "virtual" neutral point.
- 3 comparators connected to each phase voltage and the virtual neutral point to easily determine when the phase voltage has switched polarities for zero-crossing detection (ZCD); this is very important for tracking rotor position.

Within the motor drive system, the power management system handles power distribution and contains the required safety measures to protect against back energy from the motors. The main power source will be a lithium-ion battery rated at a nominal 12V, capable of outputting peak and average current values required for movement and hovering. A regenerative braking feature will store away extra power produced by the large currents and wattages that spike up from the motor's inductive nature. In specific, the power system has:

- A reverse polarity protection unit that operates using a power management chip like Analog Devices' LTC4367, which acts as a switch that closes when negative voltage is applied.
- TVS diode clamp that is placed in parallel with the battery to protect the rest of the circuit from voltage spikes resulting from the motors' inductive nature.
- Large parallel capacitor bank to store any excess energy generated from reverse voltage effects.
- Buck converter to buck down voltage of battery to a voltage usable by the microcontroller.
- State of charge estimation through battery voltage tracking and battery thermal tracking for safety reasons.

Regarding regenerative braking, as an optional addition, there are a few things that need to be changed or considered before implementing it with the rest of our electrical subsystem. In particular:

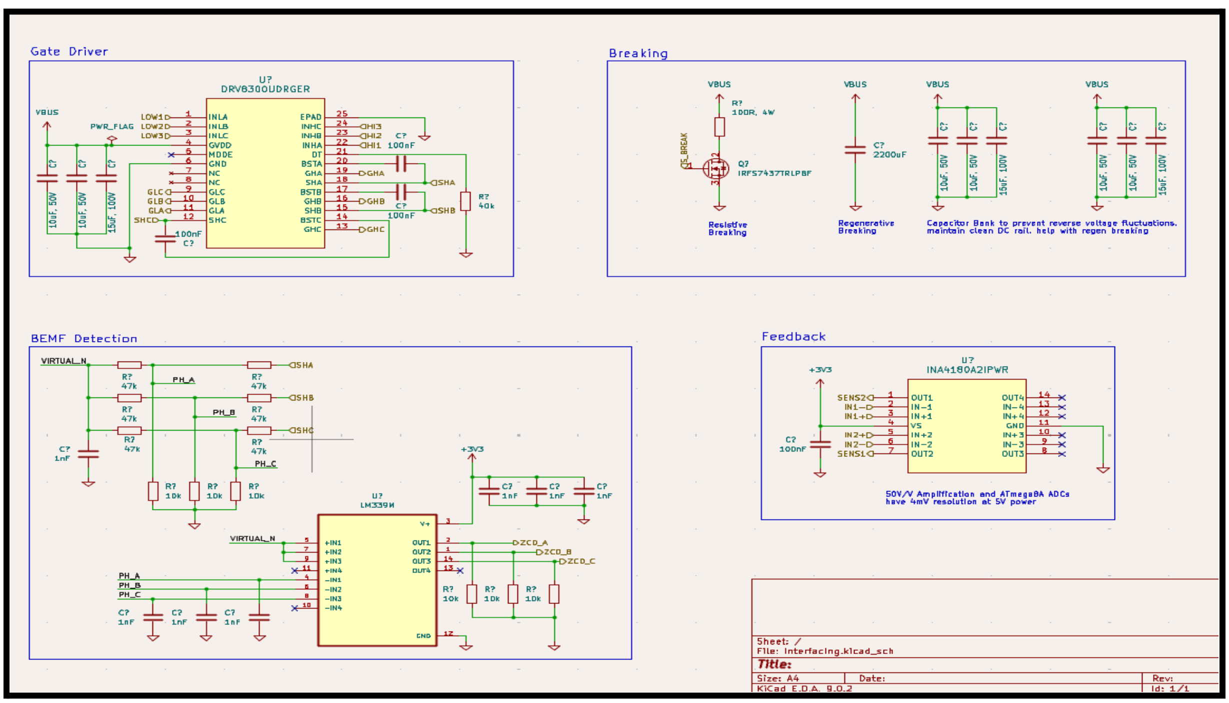
- In our case, the energy generated by each motor when slowing down is not proportional to the total weight of the drone; the motor is only connected to a singular rotor, so the energy generated is limited to being less than the energy required to spin the rotor up to its initial speed.
- In contrast to cars and trains, speeding up requires a large proportion of energy while maintaining velocity is related to frictional losses, which is relatively small unless at high velocities. For drones, the amount of energy required to spin up a single rotor is relatively small compared to the amount of energy constantly consumed to hover.
- A boost converter to increase the voltage of the reverse voltage from the motors, allowing current to flow back into the batteries (or potentially run MOSFET bridge as boost converter)

- Since all rotors will be running off the same battery packs, the battery cannot be charged and discharged simultaneously. The best solution here is to have 2 batteries in parallel, with a switching circuit that routes regenerative charging through only one battery at one time while the other battery powers the rest of the circuitry. A dedicated 3.3V battery may likely also be required for sensitive electronics that require stability.

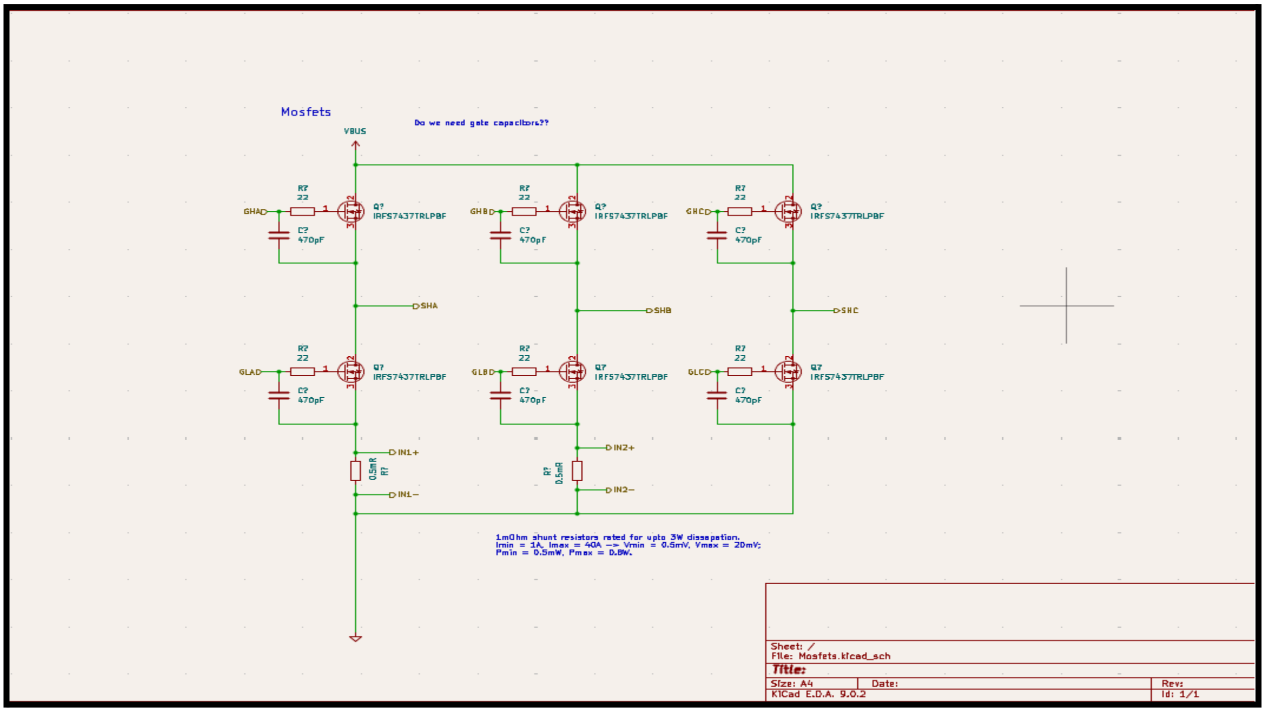
However, if using the 3-phase MOSFET as a boost converter, no extra hardware is required to externally test the feasibility of regenerative braking. If the power produced is too minimal in comparison to the complexity/instability of adding a switching circuit, we can disable the MOSFET bridge converter, and the capacitor bank and RPP will handle the reverse voltage from the motors.

Schematics

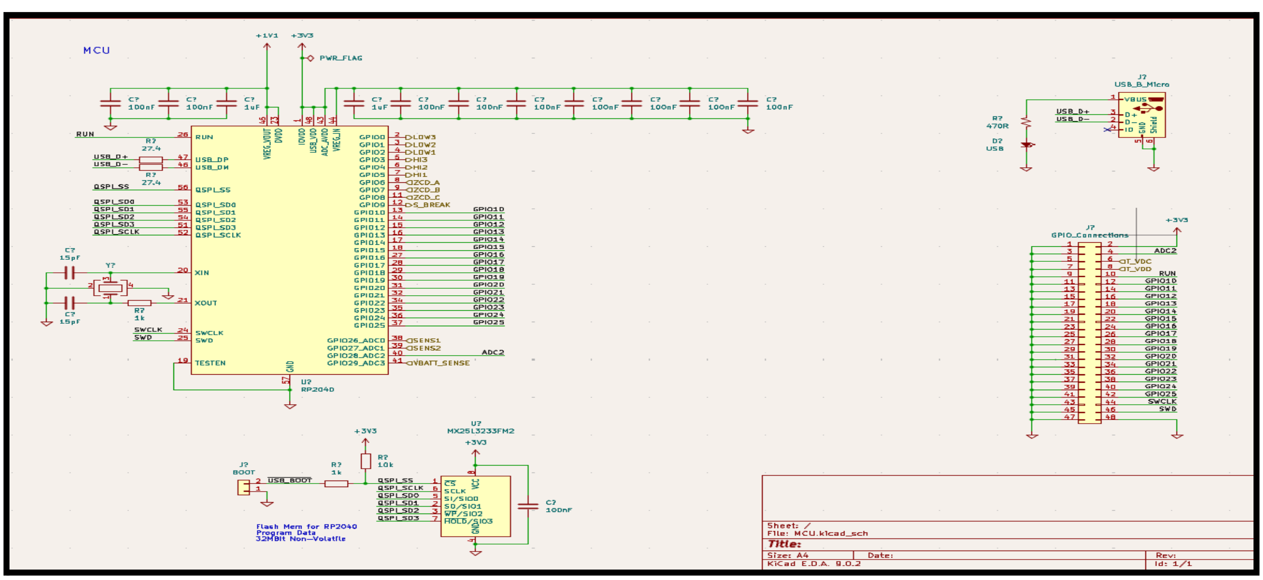
The ESC schematic can be broken into a few sections: Interfacing, MOSFET Bridge, MCU, and Power. The Interfacing circuit shown below consists of the BEMF sensing resistor network placed in parallel with each 3-phase BLDC motor, the gate driver, a capacitor bank, and sensors for phase voltage/phase current as well as total current draw.



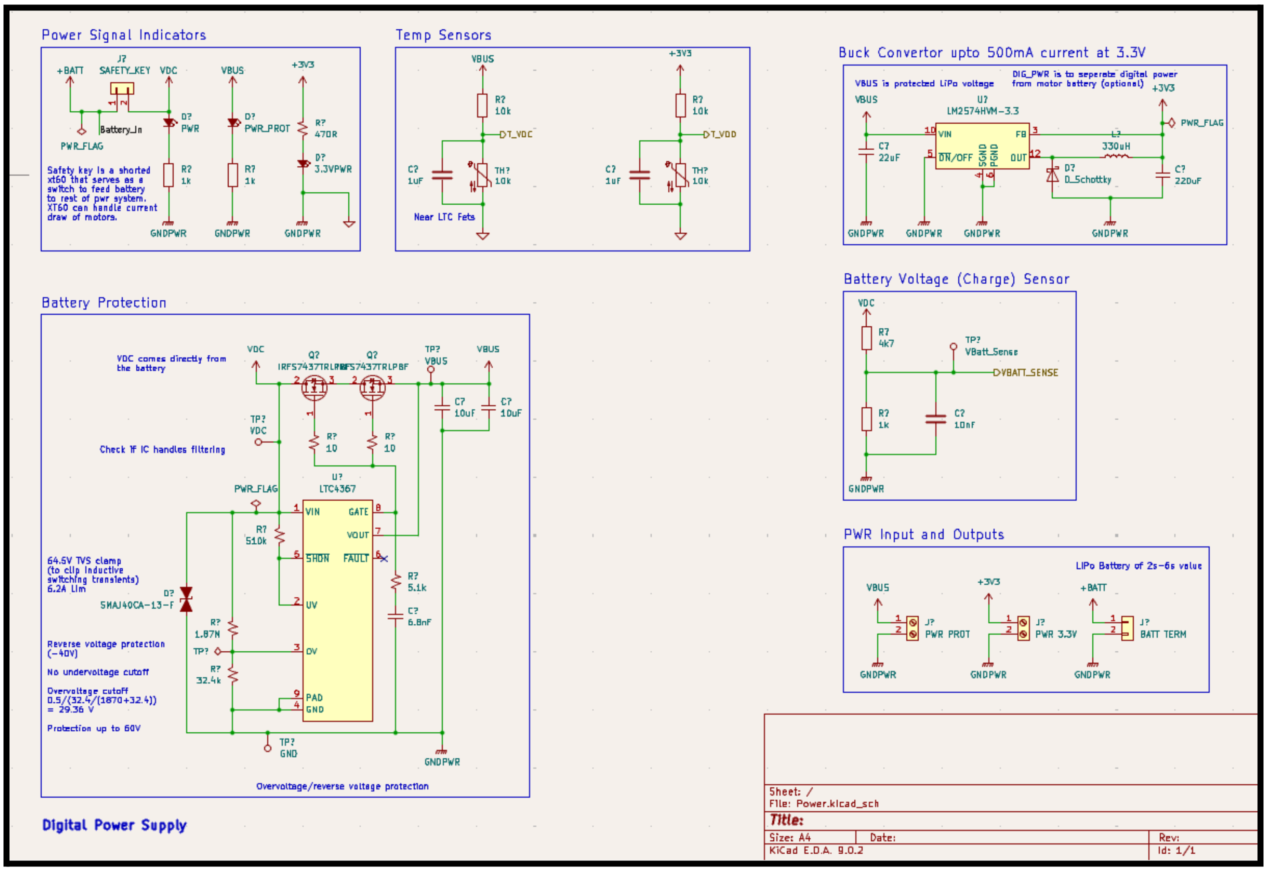
The MOSFET Bridge setup consists of several high current, high voltage MOSFETs setup in a triple half-bridge configuration, with each half-bridge powering one of the phases for one BLDC motor.



The MCU schematic consists of the RP2040 microcontroller with all necessary pull-down resistors, capacitors, GPIO connections, USB connectors, and flash memory.



The Power schematic is quite involved, and includes the temperature sensors as a failsafe, power signal indicators depending on the battery voltage, input and output ports, a buck converter to provide power to the microcontroller, and a battery protection circuit to protect the circuit from reverse polarity voltages and any transient voltage peaks.



Requirements and Verification

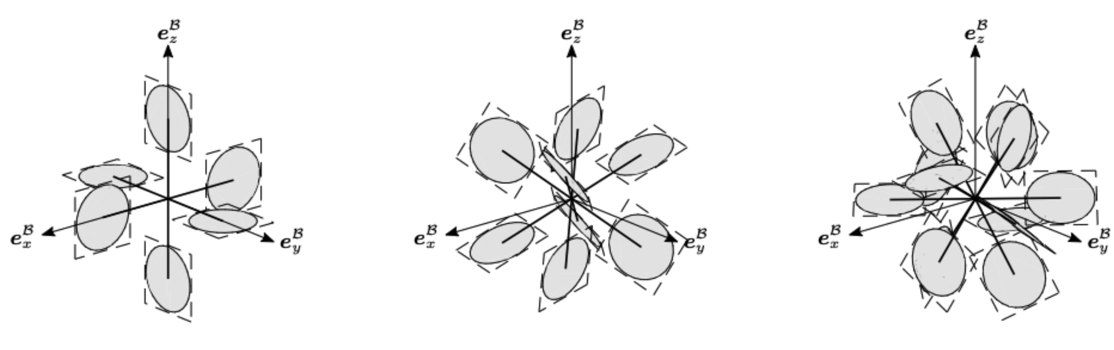
The most important requirements for this subsystem are to supply the correct voltage and current values to each of the actuators, and provide the required hardware to drive the actuators. Specific, quantifiable requirements and verifications for each requirement are listed below:

Requirement	Verification
☐ 12V battery capable of outputting at a nominal 12V ☐ 12V battery capable of power draw up to $8 \cdot 150W = 1200W$ ☐ 12V battery capable of powering flight time of around 5 minutes, approximately an energy capacity of $8 \cdot 30\ W/12V\ \cdot 1/12$ hour = 240/144 Ah = 1667 mAh	☐ **Multimeter testing** between two battery terminals, test when **fully charged** and **mostly discharged** ☐ **Current sense tracking** through **STM32 microcontroller** for each of the ESCs to verify total current draw ☐ **Discharge** fully charged battery through all 8 loaded motors until fully discharged
☐ Buck converter capable of outputting a steady 3.3V from a 12 V input ☐ Buck converter capable of current draw up to 160 mA ☐ Buck converter capable of outputting 3 different PWM channels at 100 KHz	☐ **Multimeter testing** between buck converter output terminal**s**, test when discharging battery through motors and idle ☐ **Multimeter testing** between buck converter terminals with varying resistive loads, with up to 200 mA total current draw ☐ **Oscilloscope testing** at ESC inputs to read PWM output from MCU, powered by buck converter
☐ MOSFETs bridge capable of switching at 100 KHz ☐ MOSFET bridge capable of handling VDS as high as 30V	☐ **Oscilloscope testing** at MOSFET bridge outputs to verify correct output frequency ☐ Use **power supply** to artificially supply 30V VDS to MOSFETs

	and confirm output PWM functionality
☐ Gate drivers capable of handling input PWM frequencies of 100 KHz	☐ Verify motor operation **visually** and confirm correct phase outputs for each motor phase using **oscilloscope testing**
☐ BLDC motors run at 12V	☐ **Visually confirm** motor operation with applied trapezoidal BLDC motor control algorithm

2.32 Mechanical Subsystem

The overall mechanical design focuses on the design of the frame and configurations of the motors. The frame of the omnicopter will take the form of either a 6 or 8 motor configuration depending on power draw, stability, and feasibility testing after electronics development. We're placing an emphasis on minimum weight and maximum strength while maintaining easy fabrication through quick prototyping methods like 3D printers. Regarding motor configurations, example orientations from the ETH Zurich paper for both 6, 8, and 12 motor configurations are listed below:



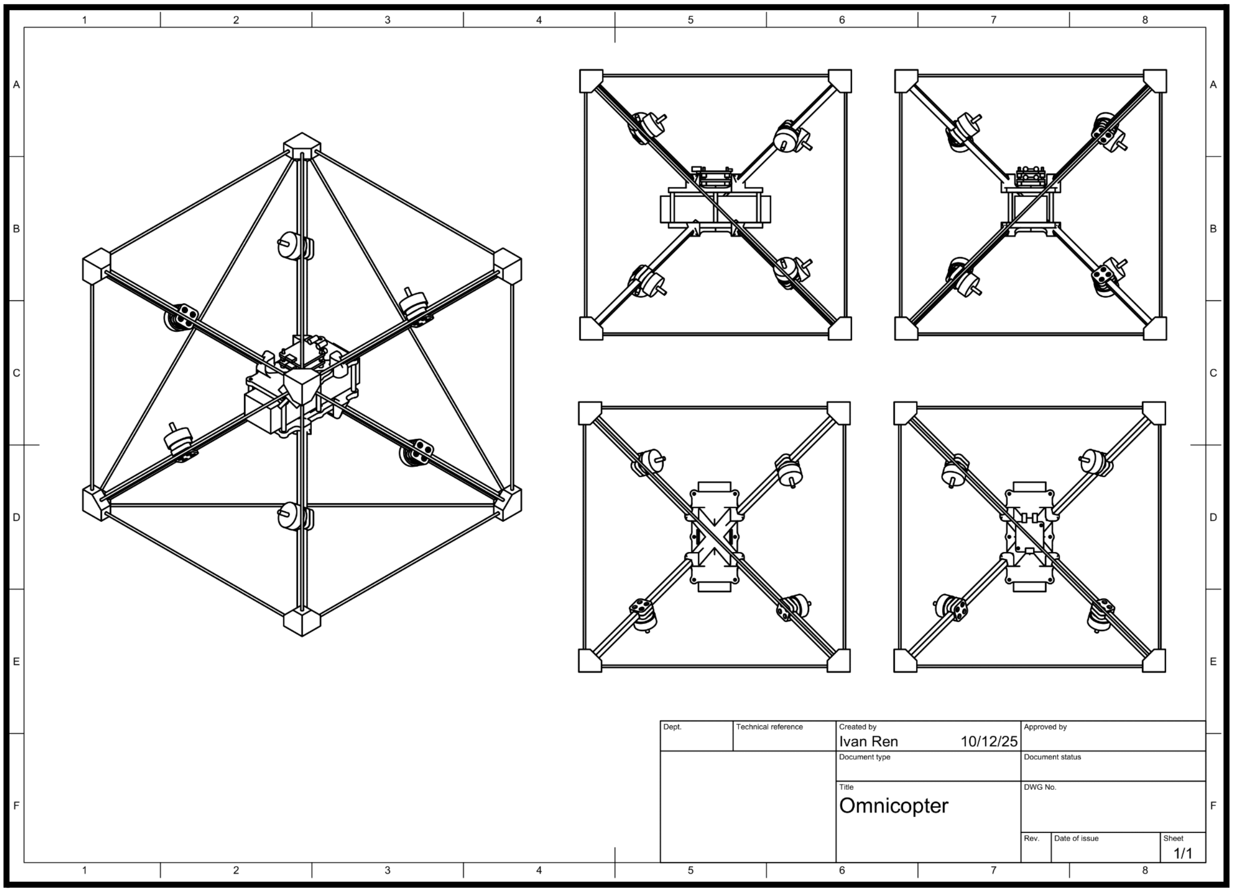
(a) $N = 6$, $r_{\max} = 1.41$ (b) $N = 8$, $r_{\max} = 2.31$ (c) $N = 12$, $r_{\max} = 4.24$

Utilizing the given dimensions and angles, an overall structure for this drone can be constructed. The frame can be split into 3 major parts:

- A central hub containing the control electronics and electrical subsystem, and 6 or 8 arms that hold each motor at a given angle. The central hub can be kept as simple as possible for weight savings. Mounting holes for all the PCBs and simple brackets to hold the batteries will be enough for the central hub.
- 6 or 8 arms, each with a mounting point for the motor, positioned at an angle that matches optimal configurations, as mentioned in the paper. For maximum weight savings, each arm will be constructed primarily from carbon fiber rods. Attachment points and connectors can be constructed out of sturdy material blends, like Tough 2000 from Formlabs' material lineup, while using medium infill percentages (30-50%) for weight savings.
- An external cage connecting each arm and providing structural rigidity to the overall frame. This cage also gives the drone some protection against crashes, as well as a landing platform. The cage can also be made of majority carbon fiber rods with 3D printed connectors.

There needs to be certain tolerances to be met when sourcing each material, which have been listed below. These tolerance measurements ensure that the drone frame remains stable and the moment of inertias around each of the axes remain constant regardless of orientation or current motion. In specific, deflection under constant loads needs to be studied to ensure that the weight of the drone and the components don't cause the carbon fiber rods to bend more than a negligible amount. Each of the 3D parts need to have a certain tolerance to ensure that the carbon rods we purchase can properly fit in joints. Motor mounting onto the carbon rods needs to be done with minimal weight usage, while also ensuring that the stress applied to the rods using screws/nuts does not fracture or damage the carbon, and that the fittings are strong enough that forces/torques generated by the motors will not create deviations in the positions of the mounted motors.

Design Documents



Requirements and Verification

Requirement	Verification
☐ Central hub should have dimensions large enough to holding 12V lithium, ESC boards, flight control + telemetry, and MCU	☐ **Physically verify** the size of each component once ESC has arrived, alongside other parts and lay out overall positioning of central hub to confirm necessary dimensions
☐ Arms long enough that rotors do not touch each other and that they	☐ **Physically verify** using calipers the leeway between propellers placed at expected motor position

don't interfere with the rotors' airstream	and each portion of the drone frame and maximize
☐ Carbon fiber skeleton has rods with structural tolerances within ±1 mm deflection from the straight line to ensure consistent structure ☐ Carbon fiber rods have structural integrity, capable of less than ±1 mm deflection from starting when under 5N of load	☐ **Manually measure** deflection from the straight line using ruler and calipers ☐ **Manually measure** deflection from the straight line using ruler, weights and stable surface.

2.33 Flight Control + Telemetry

The controls and communications side will handle reading and writing data from the drone to the remote controller, as well as converting movement signals into different motor power combinations to enable separate translational and rotational movement. In specific, any input from the remote controller will be converted to an analog signal detailing the magnitude and direction of the force we want to apply to the overall drone system. The main microcontroller on board will receive said signal through a 2.4 GHz frequency band on an open source, drone-optimized communication protocol called ExpressLRS (ELRS). In order to execute the given user input, the controller will need to take into account the current orientation of the drone and desired final position, and then convert this into PWM signals that can be directly routed to each individual ESC. To do this conversion, we will write our own custom firmware that reads motion data and motor feedback from the drone to dictate the output PWMs for each individual motor. The parts involved in this subsystem are as follows:

- 9 axis IMU to track translational motion and rotational motion

- 2.4 GHz antenna and receiver
- 2.4 GHz 8 Channel remote controller with simple dual-joystick system and 2 toggable buttons, with each joystick handling 2DOF of rotational or translational motion and the auxiliary buttons controlling the remaining 2DOF
- STM32F446 microcontroller receives signals from the remote controller and transforms it into output motor PWM signals that are sent to the gate driver on each of the ESCs.

Requirements and Verification

Requirement	Verification
☐ STM32f446ZE should stably operate using 3.3V digital input from buck converter ☐ Capable of SPI, I2C and UART communications at high bus clock rate of ~400 kHz for low latency and high sampling rate ☐ Processor clock rates of ~32MHz	☐ Verify operation of MCU by **connecting to PC** using **3.3V buck converter output** and confirming functionality ☐ Verify sampling rate by **polling data from IMU module** at specified rates and confirming correct size of data output ☐ **Verify through microcontroller datasheet**
☐ SX1280IMLTRT encoder must take digital signals and converts to 2.4GHz radio signals ☐ Frequency accuracy: ±10–20 ppm (depends on crystal) ☐ TX output power tolerance: ±1.5–2 dB ☐ RX sensitivity tolerance: ±2 dB	☐ **Verify using 2.4 GHz receiver and antenna** connected to STM32, send and read arbitrary, distinct values ☐ **Verify through datasheet**

☐ Phase noise: −100 dBc/Hz @ 100 kHz offset (typical)	
☐ Accelerometer (IMU) ☐ Zero-g offset: ±40–100 mg ☐ Sensitivity error: ±1–3% ☐ Noise density: ~100–300 μg/√Hz ☐ Gyroscope (IMU) ☐ Zero-rate offset: ±1–5 °/s ☐ Sensitivity error: ±1–3% ☐ Noise density: ~0.005–0.02 °/s/√Hz	☐ **Verify physically and through the datasheet**, accelerate IMU in constrained directions by dropping it in different orientations and record accelerations ☐ **Verify physically and through the datasheet**, test IMU at no rotation, rotate IMU using motor and confirm output readings match expected from final angular velocity and acceleration time.

2.4 Tolerance Analysis

Geometry:

The ETH Zurich team [1] determined rotor positions and orientations by framing the problem as an optimization problem, where the goal is to maximize the inner sphere radius r_{max} while enforcing symmetry and isotropy. The rotor positions were constrained to the vertices of regular polyhedrons, so that the inertia tensor is isotropic, meaning the moment of inertia stays the same across each direction. They cast the design as $\underset{P,N}{\text{maximize}} \arg\max_{r} \left\{ r : \{ v \in \mathbb{R}^6 \mid \|v\|_2 \leq r \} \subseteq \mathcal{V} \right\}$, where P is the position of the rotors, N is the disk normals, and $\mathcal{V}$ is the set of all attainable thrusts and torques. After using MATLAB's fmincon and numerically solving for optimal rotor orientation, they chose to go with a cube orientation as it was the best balance of capability and practicality. To maximize torque, they align each rotor's disk normal perpendicular to its position vector (so the torque arm is largest). But to satisfy isotropy (equal singular values), they slightly rotate the normals (e.g. π/6 about z-axis) so that the final configuration balances thrust and torque equally. The result is a set of 8 thrust vectors (the columns of N) that are evenly spread out in 3-D space and matched to the cube vertices.

Thrust:

In [1], the authors build a 6-D wrench map from rotor thrusts to vehicle torque, they then optimize geometry and report a normalized insphere radius, $r_{max} = 2.31$. Here normalized means per-rotor thrust is normalized to $f_{max} = 1$, and rotor positions lie on the unit sphere. This means, if each rotor could deliver unit thrust, the vehicle could guarantee any combined wrench $\begin{bmatrix} f \\ \tau \end{bmatrix}$ whose Euclidean norm ≤ 2.31. Real fixed-pitch motors can't go to zero, they need a minimum RPM. In [1], they handle this with $f_{max}^{eff} = f_{max} - 2f_{min}$ an "effective" per-motor bound, . Scaling this normalized radius by our per-motor limit gives us our guaranteed 6-D wrench radius, $r_{wrench} = r_{max} f_{max}^{eff}$. Using this, for our drone to be able to hover at any given direction, we are going to need to satisfy the requirement $r_{wrench} \geq S_f mg$, where S_f is a safety factor, and m is the weight of our drone. Since r_{wrench} is for force and torque, meeting this should be sufficient for the force that our motors will need to generate in order to guarantee omnidirectionality.

The motors we intend to use in the design are MRM Titan 2208-1100KV. Based on the datasheet of the motor, each motor can provide a thrust of 6.25N using 5cm rotors. When multiplying the individual thrust with the insphere radius, we get a max symmetrical thrust of 14.4375 N. This thrust provides us with a weight budget of 1473g.

Frame:

We intend to use a 3D printer along with 3D modeling software to design and fabricate the frame for our omnicopter. A critical aspect of the design is maintaining geometric symmetry. The more symmetrical the frame is, the more generalizable and predictable our control algorithms will be across different orientations of the drone. However, perfect symmetry is difficult to achieve because components such as the battery and flight controller introduce unavoidable asymmetries due to their uneven weight distribution.

Since we are using eight identical motors and ESCs, the propulsion system itself remains largely symmetrical. The main challenge lies in compensating for the mass imbalance introduced by the heavier subsystems. To address this, we deliberately orient these components in directions where the motors are capable of producing thrust well above the critical minimum required for stable flight. While some orientations of the drone inherently allow for higher maximum thrust output than others, strategically biasing the weight distribution toward these stronger thrust directions allows us to effectively calibrate the system. This approach enables the drone to behave more like a pseudo-symmetrical system, improving control performance without requiring a perfectly balanced frame.

Battery:

Based on our motors and their theoretical draw, we can expect a maximum of 1864W from the battery. We plan to use a 4s LiPo battery for our design and by using the formula P = IV, this comes out to a maximum current draw of 155A across all 8 motors or 20A per motor. The maximum current draw of a LiPo battery is given by $I_{max} = C_{rating}$ * Energy. Where the C_{rating} is a LiPo-specific parameter corresponding to the battery's current draw. The energy here is in Ah. Then, for example a 3100 mAh battery would need to be around 50C to provide the needed current. For our design, we're planning on using a 3300 mAh battery rated for 60C. This battery gives us critical leeway in potential cases where certain motors might require extra current draw to account for unbalanced weight distribution and propeller imperfections.

Weight Budget:

Our total weight budget is 1473g. This weight needs to include the battery, motors, frame and electronics. The heaviest of these components are going to be the motors and batteries. The current battery weight is 318g and total motor weight 448g. As such, we have a total electronics and frame budget of 707g. We're aiming to have a total frame and electronics weight of 500g to provide extra tolerance in weight load.

3. Cost and Schedule

3.1 Labor Analysis

We need to first consider labor costs by roughly estimating our salary according to our education and qualifications. Our project is split into 3 main portions: circuits design and electronics testing, mechanical design and fabrication, and controls/software development. Each of our roles will require a similar amount of time, considering a timeline of around 10 weeks (including planning and circuit design during the summer), we're assuming a total of 100 hours, at 10 hours per week. Considering each of these main roles, as well as our relative levels of experience, we used the following jobs titles to estimate our total cost: $5800 + $5100 + $6300 = **$17200**

Job Title (New Grad)	Hourly Wage	Hours Worked	Total Cost
Hardware Engineer	$58/hour	100 hours	$5800
Mechanical Engineer	$51/hour	100 hours	$5100
Software Engineer	$63/hour	100 hours	$6300

3.2 Cost Analysis/BOM

The second portion of our analysis focuses on the actual estimated cost of acquiring all the parts for the drone. Several parts will be purchased in larger quantities than strictly required, however this is due to the soldering process and high probability that some boards run into some issues after soldering. The PCBs are cheaper to buy in a larger quantity, so the price is somewhat amortized by buying a larger set, while also providing us extras in case of mistakes. It is also important to mention the reasoning behind purchasing some redundant components like the 4-in-1 ESCs, this is as a last resort in case of issues with our own design. Considering the importance of the ESC system, it is absolutely necessary to have a working set of ESCs to drive the 8 BLDC motors. The total cost based off of the table below is: $**<u>643.59</u>**

ICs	**Quantity**	**Cost**
RP2040CT	12	$1.22 x 12 = $14.64
MX25L3233FM2	12	$0.48 x 12 = $5.76
DRV8300UDRGER	12	$1.44 x 12 = $17.28
INA4180A2IPWR	12	$1.00 x 12 = $12.00
IRFS7437TRLPBF	96	$0.4911 × 96 = $47.07
LM2574HVM-3.3	12	$5.31 × 12 = $63.72
LTC4367	12	$7.07 × 12 = $84.84

Capacitors	**Quantity**	**Cost**
0201, 10nF, 25V	12	$0.10 x 12 = $1.20

0201, 1nF, 25V	84	\$0.10 x 84 = \$8.40
0201, 470pF, 25V	72	\$0.10 x 72 = \$7.20
0402, 100nF, 25V	168	\$0.10 x 168 = \$16.80
0402, 15pF, 50V	24	\$0.20 x 24 = \$4.80
0402, 1µF, 6.3V	48	\$0.11 x 48 = \$5.28
0402, 6.8nF, 50V	12	\$0.15 x 12 = \$1.80
0805, 10µF, 50V	72	\$0.32 x 72 = \$23.04
0805, 10µF, 25V	24	\$0.11 x 24 = \$2.64
0805, 22µF, 10V	12	\$0.32 x 12 = \$3.84
0805, 220µF, 25V	12	\$0.91 x 12 = \$10.92
0805, 2200µF, 25V	12	\$1.97 x 12 = \$23.64

Resistors	**Quantity**	**Cost**
0402, 47kΩ, 0.1W	72	\$0.10 x 72 = \$7.20
0402, 1kΩ, 0.1W	60	\$0.10 x 60 = \$6.00

0402, 27.4Ω, 0.1W	24	\$0.10 x 24 = \$2.40
0402, 22Ω, 0.1W	72	\$0.10 x 72 = \$7.20
0402, 10kΩ, 0.1W	132	\$0.10 x 132 = \$13.20
0402, 1.87MΩ, 0.1W	12	\$0.10 x 12 = \$1.20
0402, 32.4kΩ, 0.1W	12	\$0.10 x 12 = \$1.20
0402, 470Ω, 0.1W	24	\$0.10 x 24 = \$2.40
0402, 510kΩ, 0.1W	12	\$0.10 x 12 = \$1.20
0402, 10Ω, 0.1W	24	\$0.10 x 24 = \$2.40
0402, 5.1kΩ, 0.1W	12	\$0.10 x 12 = \$1.20

Miscellaneous (ESC)	**Quantity**	**Cost**
STM32F446	2	\$8.63 x 2 = \$17.26
MPU-9250	2	\$6.78 × 2 = \$13.56
SX1280IMLTRT	2	\$6.45 x 2 = \$12.90

Drone	Quantity	Cost
MRM Titan 2208-1100KV	8	$7.50 x 8 = $60.00
Thunder Power RC TP1800-4SM70	1	$53.99 x 1 = $53.99
Carbon Fiber Rods	15	$10.99 x 3 = $32.97

Controller	Quantity	Cost
RadioLink 2.4 GHz 8Ch Controller with 2 Joysticks	1	$52.44 x 1 = $52.44

3.3 Schedule

Due to the complexity of the ESC and the overall complexity of the entire project, we have a few deadlines that we plan to make ahead of the official turn in dates. This is to push our hardware testing and validation forward in order to give our software development team enough time to begin developing the controls and algorithms we need to control the drone. This also allows us multiple rounds to send off new PCB designs to manufacturing in case we find errors while testing. Once we finish hardware, this will also allow us to remotely focus on software development over Fall Break.

Week	Tasks

Week of 9/29	• Validate ESC PCB, PCB Review • Purchase required remote controller and MCU + IMU + 2.4 GHz receiver • Meet with Jack Blevins for extra guidance and design verification • Finalize timeline
Week of 10/6	• Send off finalized ESC PCB • Purchase required ESC PCB components from DigiKey • Begin in-depth software research on controls • Breadboard flight control with STM proto-board • Finalize design document
Week of 10/13	• Purchase all other required parts • Assemble first version of ESC PCB ○ Test using STM32 proto-board and power supplies • (Optional) Design custom PCB for flight control/telemetry • Make necessary design changes, prepare to send out edited design in second round of PCB orders • Contact Machine Shop once dimensions finalized, carbon fiber rod cutting may be necessary • Write out flight control algorithms to test on breadboarded flight control/telemetry

Week of 10/20	● Test second version of ESC PCB ○ If necessary, edit and send out third version ○ Otherwise, assemble all other 8 boards + extras ● (Optional) Finalize flight control/telemetry board ● Begin ESC + BLDC motor algorithm development (motor start up, speed up, slow down, etc) ● Contact surrounding labs with drone flight facilities for facility access
Week of 10/27	● Verify proper motor function when running all 8 BLDC simultaneously off of power supply and direct connection to STM32 proto-board ● Confirm stability when holding speed, ramping up, slowing down ● Confirm temperature limits are met, probe boards for potential damage/overvoltage/overcurrent ● Begin physical design process ● Cut carbon fiber rods to size
Week of 11/3	● Confirm motor functionality when attached to battery instead of power supply ● Confirm motor functionality with signals from remote controller instead of simulated on STM32 ● Manufacture required parts for

	drone frame, begin assembly

Week of 11/10	• Attach motors, flight control, battery to frame
	• Begin flight testing, focus on achieving lift off
	  ○ Use power supply if necessary to remove battery weight
	  ○ Confirm drone is capable of moving up in any capacity
	• Check for mechanical defects, redesign and remanufacture

Week of 11/17	• Continue flight testing, focus on steady hovering in starting orientation ○ If achieved, attach battery and retest • Ensure all electronics are still operating within range, no damage, etc
Week of 11/24 (FALL BREAK)	• Remote software development • Movement tuning depending on test results, compensate for center of mass differences, etc • Develop more motion controls, focus on translational movement
Week of 12/1	• Continue flight testing, focus on translational movement in starting orientation ○ Attempt orientation change using rotational changes • Final demos!

4. Ethics and Safety

Public and Operator Safety

Our omni-directional drone features high-speed rotating propellers and a Li-ion battery pack, which pose risks of injury, fire, or electrical failure. To mitigate these risks, all testing will initially be conducted in restricted environments such as netted indoor flight spaces or designated outdoor zones, in line with University of Illinois Laboratory Safety Protocols for Unmanned Aircraft Systems (UAS). Early flights will use tethers to reduce potential crash energy, and propeller guards will be installed during development. An emergency kill switch will be implemented to immediately cut motor power in case of malfunction.

Electrical and Battery Safety

Lithium-ion batteries can overheat or catch fire if improperly charged or discharged. We will follow the U.S. Consumer Product Safety Commission (CPSC) guidelines for lithium-ion batteries and adhere to UIUC's Electrical and Computer Engineering (ECE) lab policies on handling rechargeable cells. Specific measures include reverse polarity protection, fuses, TVS diodes, and capacitor banks to absorb back-EMF from motors. Charging will be supervised using manufacturer-recommended chargers, batteries will be stored in fire-retardant containers, and regular inspections will be documented to monitor swelling or damage.

Mechanical Integrity and Pre-Flight Checks

Drone crashes present hazards to people and property. Following UIUC Drone Safety Guidelines and general UAS operational best practices, each flight will be preceded by a structured inspection protocol: ensuring frame integrity, checking for loose screws or connectors, and verifying propeller attachment. MOSFET and motor thermal checks will be conducted during operation to prevent overheating. These procedures will be consolidated into a written Safety Manual that details pre-flight inspection, propeller installation, battery handling, and emergency response.

Ethical Use and Privacy

We recognize that drones can be misused in ways that compromise privacy or disturb the environment. Following the ACM Code of Ethics, sections 1.2 (Avoid Harm) and 1.6 (Respect Privacy), we commit to using this vehicle only for academic and research purposes. No visual or audio data unrelated to flight performance will be collected, and we will responsibly recycle or dispose of all batteries and electronic waste to minimize environmental harm.

RF Exposure and Communication Safety

Our project uses a 2.4 GHz wireless remote controller and telemetry module to transmit and receive flight commands. These components fall under unlicensed operation limits defined by the FCC Part 15 Subpart C rules for intentional radiators in the 2.4 GHz ISM band. To ensure compliance, we will use only FCC-certified transceiver modules that meet the maximum effective isotropic radiated power (EIRP) limits ($\leq$ 1 W) and comply with specific absorption rate (SAR) limits for RF exposure. Since the modules are low-power consumer-grade devices designed for hobbyist and research use, the expected RF exposure is well below harmful thresholds.

Interference is another consideration, as 2.4 GHz is a crowded spectrum used by Wi-Fi, Bluetooth, and other wireless devices. We will mitigate this risk by operating in designated lab or field test areas where wireless traffic is controlled, and by following UIUC Laboratory UAS Operation Guidelines to avoid interference with nearby research equipment. Should interference or connectivity issues arise, testing will be halted until safe communication is reestablished.

Fabrication Safety (3D Printing, Soldering, and Machining)

When working with 3D printing, solder, and carbon fiber, there are specific safeguards that need to be taken to avoid accidental ingestion of harmful materials. Additively manufactured parts, such as those made from nylon or resin, can release fine particulates during post-processing, therefore sanding or trimming should be performed with gloves and masks in accordance with UIUC Laboratory Safety Guidelines for Additive Manufacturing. Additionally, certain 3D filaments like ABS will be avoided due their release of toxic fumes when printing; instead, alternatives like PETG or proprietary material blends that are certified safe will be utilized in our SLA and FDM printers. Soldering exposes operators to flux fumes and molten metal, therefore all soldering will be performed in well-ventilated areas or under fume extraction hoods, and eye protection will be worn to prevent burns from solder splatter.

Carbon fiber machining introduces the risk of inhaling fine dust that is harmful to lungs and skin. Cutting or sanding carbon fiber will be done in controlled environments with protective equipment, including N95 (or higher-rated) respirators, safety goggles, and gloves, following OSHA guidelines on composite material handling. All debris will be collected with HEPA-filter vacuums and properly disposed of to prevent secondary exposure.

For general machining (e.g., drilling or cutting aluminum brackets), we will comply with the Grainger College of Engineering machine shop safety rules, which require training, the use of personal protective equipment (PPE) such as safety glasses and closed-toe shoes, and never operating machines unattended.

5. References

[1] E. Ackerman, "ETH Zurich's Omnicopter Plays Fetch," *IEEE Spectrum*, May 17, 2017. [Online]. Available: https://spectrum.ieee.org/eth-zurich-omnicopter-plays-fetch. [Accessed: Sep. 18, 2025].

[2] B. Brescianini and R. D'Andrea, "An Omni-Directional Multirotor Vehicle," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 903–910, Apr. 2018.

[3] DeepBluEmbedded, "STM32 ESC PCB Design / FOC ESC (BLDC) Schematic." [Online]. Available: https://deepbluembedded.com/stm32-esc-pcb-design-foc-esc-bldc-schematic/. [Accessed: Sep. 18, 2025].

[4] M. Hein, *Demystifying BLDC Motor Commutation: Trap, Sine, & FOC*, Texas Instruments, Apr. 2020. [Online]. Available: https://www.ti.com/lit/ml/slyp711/slyp711.pdf. [Accessed: Sep. 18, 2025].

[5] Jon's Workshop, "Synchronous Rectifiers," Nov. 2017. [Online]. Available: https://www.jons-workshop.com/syncrec.html. [Accessed: Sep. 18, 2025].

[6] Texas Instruments, "Integrated intelligence part 3: Motor startup from standstill position," SSZT853, Dec. 2017. [Online]. Available: https://www.ti.com/lit/ta/sszt853/sszt853.pdf. [Accessed: Sep. 18, 2025].

[7] D. Torres and P. Heath, *Regenerative Braking of BLDC Motors*, Microchip Technology Inc. [Online]. Available:

https://ww1.microchip.com/downloads/en/devicedoc/regenerative%20braking%20of%20bldc%20motors.pdf. [Accessed: Sep. 18, 2025].

[8] R. Tylor, “Minecraft, MCPE | How to make a Working Drone | No Commands, No Addons,” *YouTube*, Sep. 2, 2020. [Online]. Available: https://www.youtube.com/watch?v=FzrqVVbJ7EI. [Accessed: Sep. 18, 2025].

[9] Semtech, *SX1280/SX1281 — Data Sheet*, Rev. 3.2, Mar. 2020. [Online]. Available: https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/483/SX1280-81_Rev3.2_Mar2020.pdf. [Accessed: Sep. 18, 2025].

[10] “Omnicopter | PX4 Guide,” *PX4 Documentation*, [Online]. Available: https://docs.px4.io/main/en/frames_multicopter/omnicopter.html. [Accessed: Oct. 9, 2025].