Creative Engineering - The Drone Project

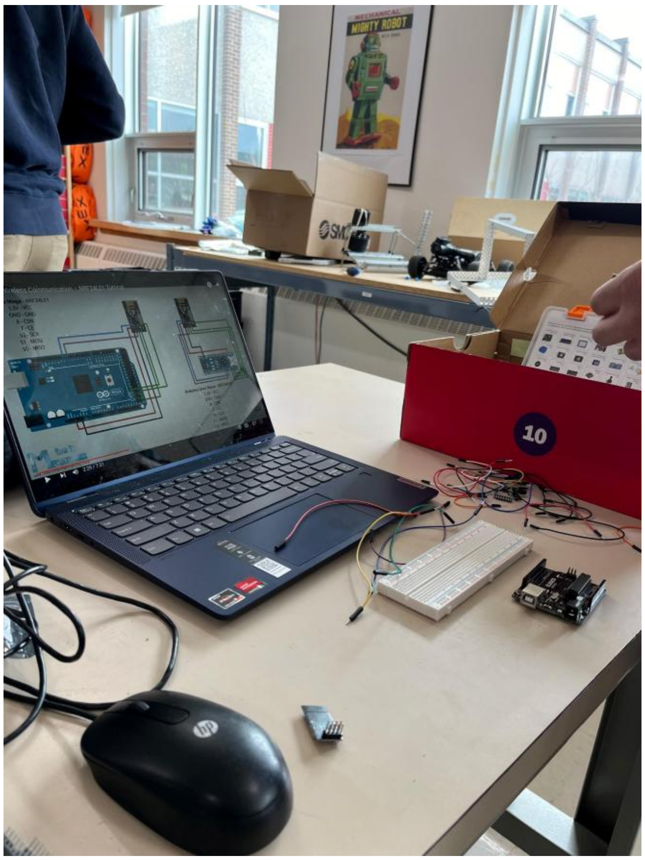
Saahil Patel | Logan Gillett

Hey! We're Saahil and Logan, and we've decided to make a drone for our Advanced Engineering class. We've recorded which websites we used for research purposes on this spreadsheet and tracked the project's expenses, which can be found here.

Drone Images & Documentation

Electrical - Designing and Building Circuits Initial Research - Using the nRF24l01:

-- Wiring the nRF24l01 With an Arduino Uno --



Initial Construction:

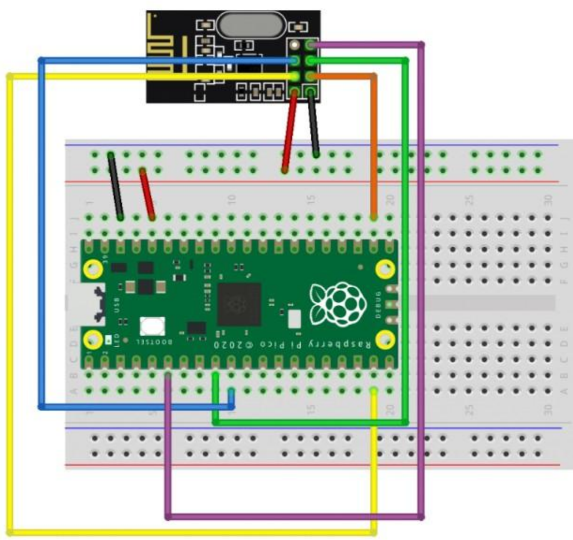
Initially, we tested constructing the drone using two Arduino Unos and sending joystick inputs to the drone to control its movement. We experimented with two nRF24l01s in the first few days, using the Unos to control sent and received information. After considering size constraints and voltage limitations, we decided to use two Raspberry Pi Picos for the final design, allowing for a more compact way of sending the information to the drone.

-- Rewiring With the nRF24s to Work With Raspberry Pi Picos --

Switching to Picos:
Following diagrams that we found in an article about wiring an nRF24l01 module to a Raspberry Pi Pico, we created our first circuits using the nRF24.

-- Wiring Diagram --

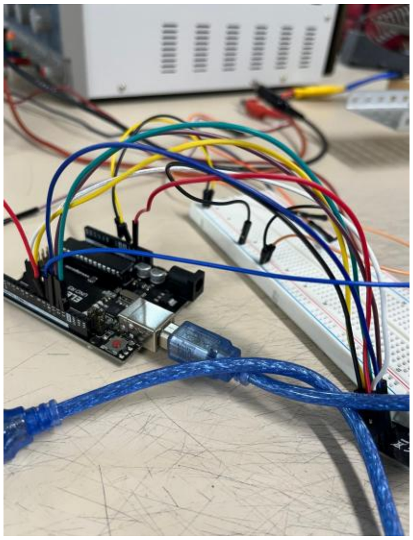
-- Wiring Table --



Pico	NRF24L01+
Pin 36 / 3V3 Out	VCC
Pin 38 / GND (or any other GND Pin)	GND
Pin 22 / GP17	CE
Pin 19 / GP14	CS
Pin 9 / GP6	SCK
Pin 10 / GP7	MOSI
Pin 6 / GP4	MISO

The receiver and transceiver are wired identically and declared separately in their codes respectively. The nRF24l01 is wired as follows: the VCC (red) pin is connected to pin 36 on the pico, powering it with 3.3V, while the ground wire (black) is connected to pin 38, acting as a ground for the component. The CSN (CS) pin is connected via the yellow wire and is a digital pin that controls whether the nRF is active or not, and the CE pin, connected to pin 22 through the orange wire, controls what data is sent to and from the transceiver and receiver. The green wire connects the SCK pin, controlling an internal clock within the nRF, and the blue wire connects to the MOSI (master out, slave in) pin, which controls the microcontroller and the 24l01. The microcontroller acts as the master while the 24l01 acts as the slave where the slave sends data using the microcontroller. Finally, the MISO (purple wire) is the opposite of the MOSI where the microcontroller (master) goes in and the 24l01 (slave) goes out.

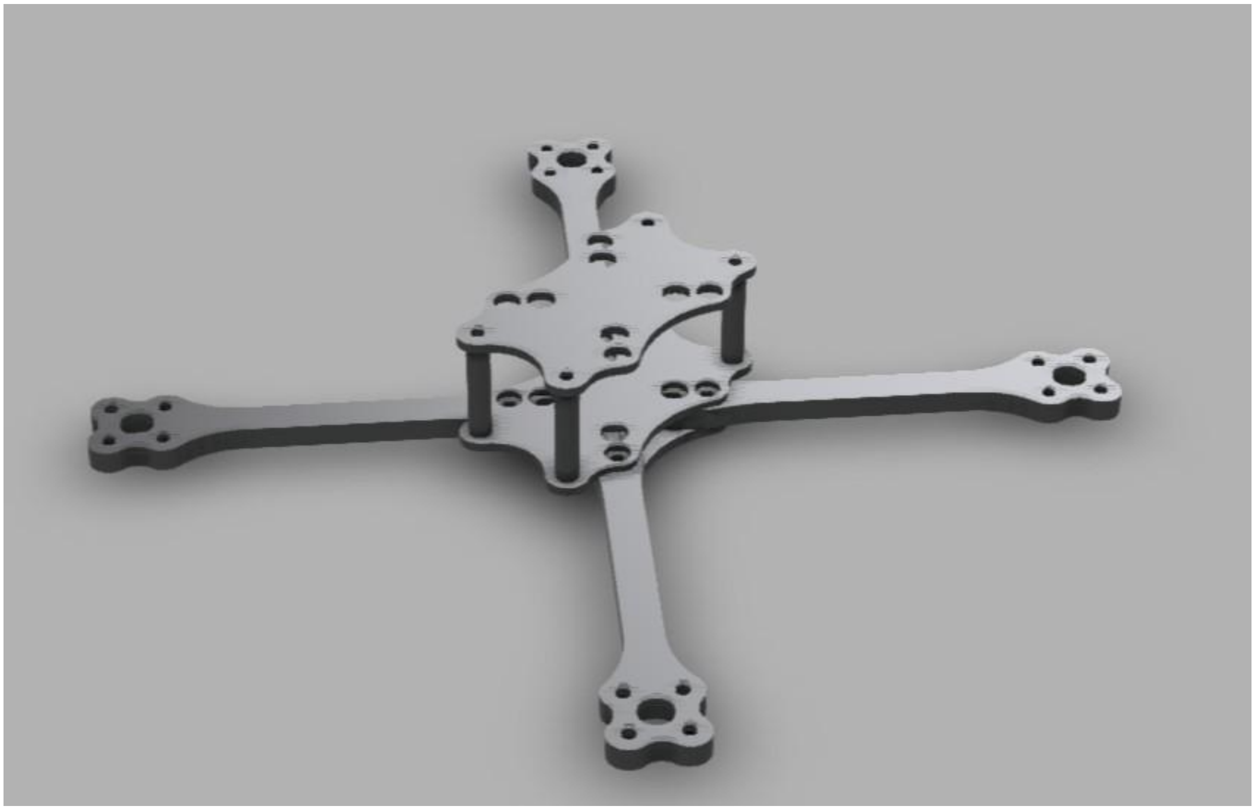
-- Revaluation --



Finding Success With Arduino Unos:

After attempting to wire the motors using Raspberry Pi Picos, we had issues making the nRF24s send data between circuits. Without success sending even one message, we decided to switch back to what Arduino Unos as we found success with them, and after a short amount of time testing, we managed to send data.

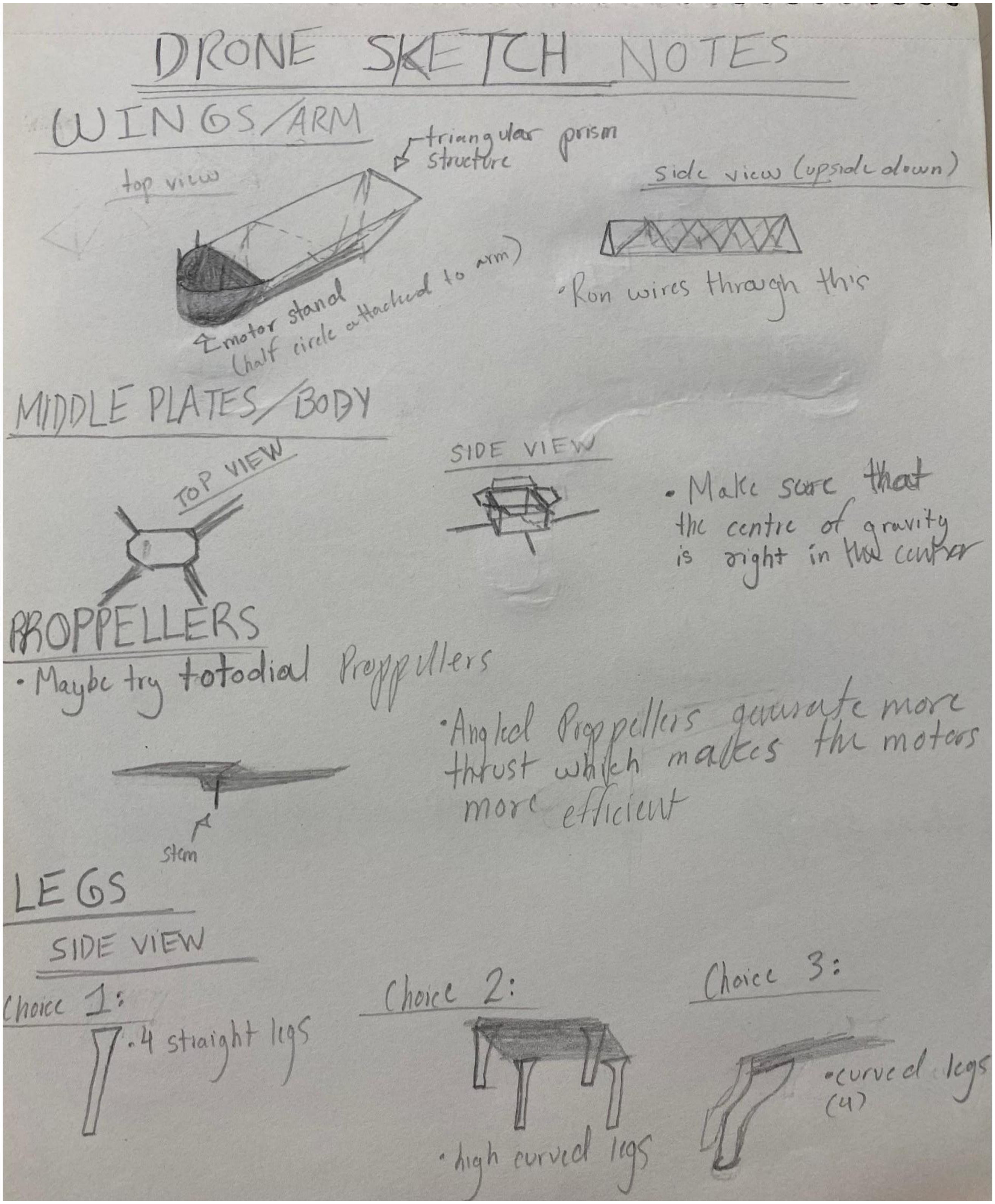
CADing - Designing the Drone Build:

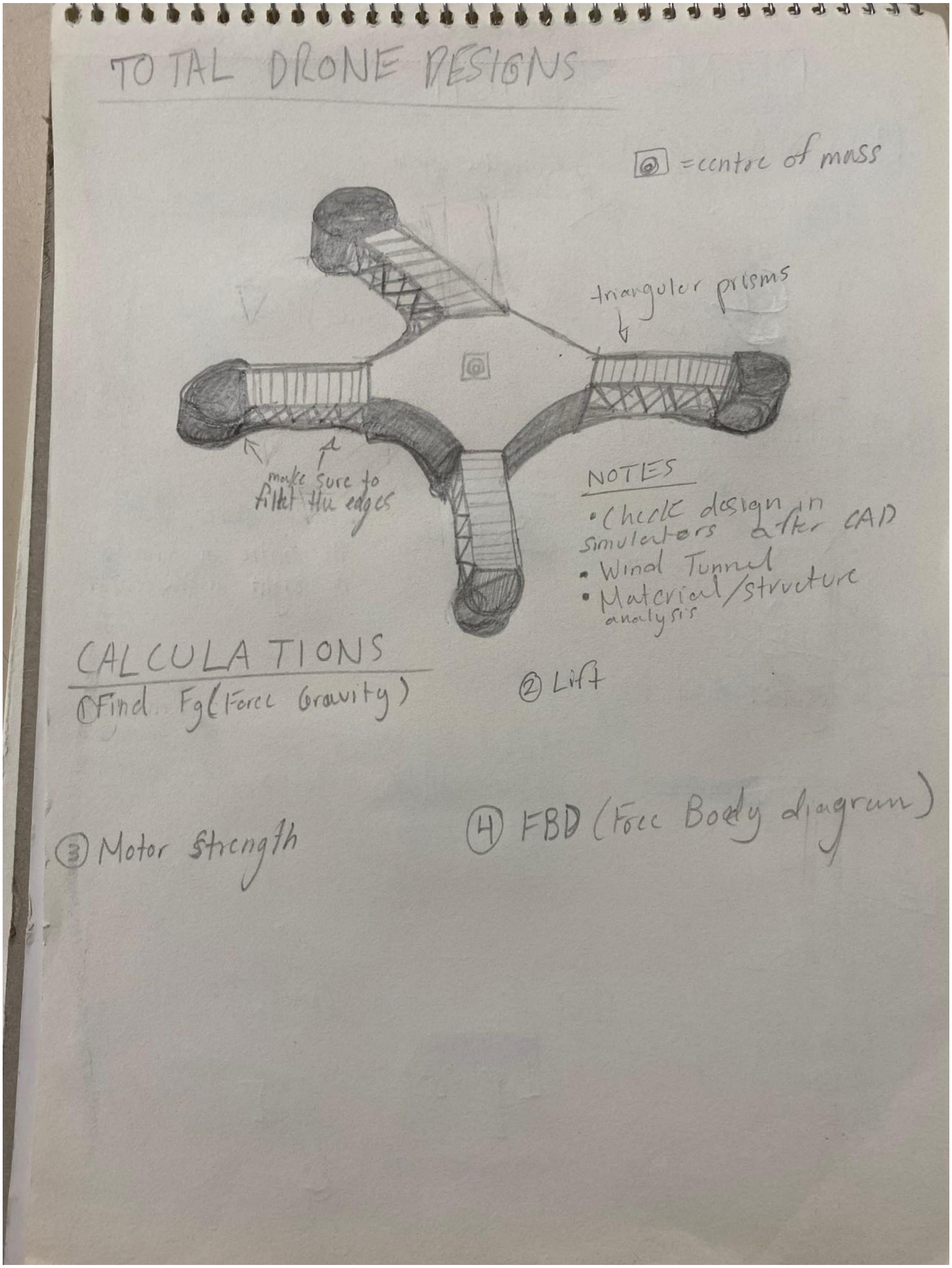


This is the V1(version 1) for the drone CAD. We used a tutorial to learn how to CAD a drone properly before we made our final versions. We learned quite a bit about CAD using this mini-project:

1. How to use vector tools like coincident, coplanar, perpendicular, collinear, and tangent
2. How to use parameters for a versatile design process
3. Efficient ways to create objects and sketches (CAD practices)
    a. Distance tools
    b. Fillet
    c. Creating efficient, solid objects with minimal constructors
        i. No useless lines
        ii. No embedded object
        iii. No overlapping objects

The following images are sketches of our version 2. This aimed to get new ideas out before we did version 2.


DRONE SKETCH NOTES
WINGS/ARM
top view
triangular prism structure
side view (upside down)
motor stand (half circle attached to arm)
•Ron wires through this
MIDDLE PLATES/BODY
TOP VIEW
SIDE VIEW
•Make sure that the centre of gravity is right in the center
PROPPELLERS
•Maybe try totodial Proppullers
•Angled Proppellers guarate more thrust which makes the motors more efficient
stem
LEGS
SIDE VIEW
Choice 1:
•4 straight legs
Choice 2:
•high curved legs
Choice 3:
•curved legs (4)


TOTAL DRONE DESIGNS
= centre of mass
triangular prisms
make sure to fillet the edges
NOTES
• Check design in simulators after CAD
• Wind Tunnel
• Material/structure analysis
CALCULATIONS
① Find Fg (Force Gravity)
② Lift
③ Motor Strength
④ FBD (Force Body diagram)

Milestones - First Reflection:

What observations (during the Research phase) have you made in planning your system?

The main observation in our research phase was made during our time speaking with a mechanical engineer at Curtis Wright who taught us the importance of structural analysis and how that can help turn brittle, weak materials into stronger structures for aerial applications. The idea he taught was Truss Analysis, a concept we are using to CAD lighter-weight, stronger arms that are hollow with trussing instead of a solid bar connecting the motors to the body. This construction method allows us to use materials like weak PLA, turning them into less fragile, more usable materials. After looking at many drone designs, we realized that some drones fail because of weak and heavy structures, and some drones may lose a lot of power and efficiency when flying due to this.

List your Essential and Non-Essential Requirements. Which ones will be the most difficult to meet?

Our essential requirements are:

1. Getting nrf24l01 transceivers working so we can transmit and receive information throughout the entirety of the project
    a. Used for motors
    b. Used for a camera (Listed in Extra Requirements)
    c. Used Flight computer (Listed in Extra Requirements)
2. Getting brushless outrunner motors working with ESC (Electronic Speed Controller)
    a. Getting this working with some sort of battery/power source
3. Getting the final CAD so we can get our design built
4. Getting a basic controller working with the radio transmitter to control air flight
5. Getting Raspberry Pi Pico H working with all of its components
    a. Soldering and running wires through drone arms
6. Programming a basic flight system

Our non-essential requirements are:

1. Getting a Flight Controller or Flight Computer for accurate flight movement and POTENTIALLY programming the drone to be autonomous
2. Camera to detect objects
3. Creating a flight PID controller

Essential Requirements:
The most difficult tasks in the essential requirements section would be programming a basic flight system and getting the transceivers working. Regarding the other components, we have worked with H-bridges and motors in the past so we don't think that part on its own should cause much trouble.

Non-Essentials:
A very difficult component of the non-essential requirements section would be programming a PID system. The MOST difficult part of the whole project would be making the drone autonomous. This would take a lot of time, research, advanced math, advanced programming, and LOTS of testing.

What do you think will be your biggest challenge in trying to succeed in this project? Think about things like time constraints, budget, expertise, and resources.

Time: We have around 2.5 months for this project which seems like a lot of time, but it isn't. It will be a challenge for us to ship materials and print/build a frame, considering there is a lot of work to do and many things can break (i.e. The drone frame). Solutions to this include minor cuts in expectation, taking the drone home to work on circuits, and buying components like motors in advance.

Budget: DRONES ARE EXPENSIVE. As a result, we recognize that we may have to do some weight-saving and proper, lightweight build techniques to account for this. We plan on covering the cost together, as we are working in a pair, which should lower the individual costs.

Expertise: The goals we have are very ambitious and difficult to achieve but we can make lots of progress and learn lots. We aren't experts in this area but that's the whole point of doing the project. We are very eager to learn online, as well as from professional engineers that we have connections with.

Resources: We don't think resources will be a major concern for the project. Many of the components are already in the classroom or just a day's ship away from Amazon, so testing with what we need won't be much of a problem. For example, Evan has kindly let us borrow 2 transceivers so we can test early and see if they are what we are looking for. With this opportunity, we learned early into the project how to program and incorporate nRF24s with a Raspberry Pi Pico H.

Productivity: This project is a major undertaking, therefore we understand that we should set realistic goals for ourselves every week, taking on small chunks of the project slowly. On top of this, keeping clean documentation should allow for a more focused mindset on the next steps. In the last week, our goals were to send and receive information through the nRF24s and CAD a basic model of the drone. After finishing the first CAD design, we received extensive feedback from a mechanical engineer on the model, which, from that, we used to create sketches of possible new iterations of the design. This week, we are almost finished getting the transceivers to send information back and forth, and we have plans to create a version 2 CAD shortly that incorporates the Raspberry Pi Pico and the propellers as well as what we've learned regarding truss analysis.

Project Damage (Drone or other components are likely to break): We have to be very careful when considering risk. We can't just test fly as soon as we attach propellers, just in case something goes wrong, so we are going to test our code and hardware separately from the drone before moving it to the drone itself and testing with restricted movement (Taping it to a pole and then testing motors and controls). Research has shown that many people have tried this before, and from their tests, it seems like a very effective and safe way to test the drone's durability without risking destroying it. The concern is that, if something breaks, we will lose productivity and resources, our budget will take a hit, and our timeline will also be ruined, therefore we have to make sure to iterate a lot and test safely and efficiently.

Celebrate your successes so far. Have you developed a new design, build, programming, prototyping, or testing skills?

New CAD Design: We used version 1 CAD to learn how to design a drone in Fusion 360 properly before we made our final versions. We learned quite a bit about CAD using this mini-project:

1. How to use vector tools like coincident, coplanar, perpendicular, collinear, and tangent
2. How to use parameters for a versatile design process
3. Efficient ways to create objects and sketches (CAD practices)
    d. Distance tools
    e. Fillet
    f. Creating efficient, solid objects with minimal constructors
        i. No useless lines
        ii. No embedded object
        iii. No overlapping objects
4. There are always basic tools that we have to review like fillet and extrude

Programming Skills: We have been figuring out the transceivers to relay information from the remote control to the drone using Raspberry Pi Picos and nrf24l01(radio). We got it to send data but it is currently not being picked up by the receiver. We have spent some time troubleshooting this but now we have found 2 people (Devon and Nate) who can help us as they have gotten theirs working using similar components. This weekend we are going to order the motors so that we can start to work with those as soon as we finish with the transceivers.

Embrace your failures so far. What have you learned through mistakes and failures?

Transceiver: Currently, the receiver isn't working as intended so we plan to use code from people who have gotten it working.

Version 1 CAD: Our first CAD design looked good, but after speaking to a mechanical engineer who was worried that our drone may be too large, we found that it would be far too heavy to support itself in the air. Another major concern was that it would be too weak based on the material we used, and the type of PLA we have in class would be fairly weak for our application. Our workaround to this was to make the drone a hybrid, meaning that certain parts that don't take most of the force would be printed, while other parts could be a stronger, potentially heavier material. To help with the force, he suggested that we look into some structural analysis, mainly Truss techniques to minimize forces that could potentially damage the drone in a way that could set us back far (Mentions more in paragraph 1 of milestone report), as well as advising us to take a first principles approach to the drone.

The idea behind this approach is to boil things down to their very basics, at which point you can innovate and optimize parts of your project. After some research, we noticed that this method is very useful for project-based learning. We can boil down our drone build into the materials, size, weight, and basic applications to figure out how light, efficient, and cheap we can get our motors and other components. It is the idea of taking basic approaches to complex problems to find unique solutions.

Your Self-Assessment

Reflect on your progress **so far**. Assess yourself using the rubric below by filling in either Green, Yellow or Red.

Green - Working Above Expectations and Peers
Yellow - Meeting Expectations
Red - Struggling to Meet Expectations

Struggling to Meet Expectations	Meeting Expectations	Working Above Expectations and Peers

Content: Programming & Design
 	
☐ My program **doesn't work**
☐ My code is hard to read due to poor…
   ☐ **Spacing**
   ☐ **Indenting**
   ☐ **Comments**
☐ My code uses few programming techniques
   ☐ **Loops**
   ☐ **Variables** with appropriate data types
   ☐ **Functions**
	Content: Programming & Design
	
	☐ I can use **sensors and actuators** we learned in class and control them via code
	☐ My code is readable
	   ☐ **Spacing**
	   ☐ **Indenting**
	   ☐ **Comments**
	☐ My code uses common programming constructs
	   ☐ **Loops**
	   ☐ **Variables**
	   ☐ **Decision Statements**
	   ☐ **Logical operations**
	   ☐ **Functions**
		Content: Programming & Design
		
		☑ I can use **sensors and actuators** beyond what we studied in class and control them via code
		☑ My code is readable
		   ☑ **Spacing**
		   ☑ **Indenting**
		   ☑ **Comments**
		☑ My code uses common programming constructs, plus **advanced features** I learned on my own

Actions: Build Quality, Performance ☐ I struggle to design, build and program systems to meet the **challenge requirements** ☐ I struggle to use **sensors** and **actuators**	**Actions: Build Quality, Performance** ☐ I can design, build and program systems to meet the **challenge requirements** ☐ I can use appropriate **sensors** and **actuators** to help my system *function*	**Actions: Build Quality, Performance** ☑ I can design, build and program systems to exceed the **challenge requirements** ☑ I can use appropriate **sensors** and **actuators** to obtain *reliable results*

Process: Strategy and Character
 	
☐ Who needs a plan or a **process**?
☐ I am struggling to **persevere** and to not give up.
☐ I wish my classmate(s) weren't here
☐ Setbacks or **difficulties throw me** off and I struggle to cope with them.
☐ I **struggle to apply** what I have learned
☐ I don't really pay attention to due dates
☐ I don't take time to **test** and **evaluate** my work
☐ I struggle to **troubleshoot**
☐ I **don't reflect** on my work
	Process: Strategy and Character
	
	☐ I considered using the **Engineering Design Process**, but I am smarter than engineers around the world and decided I could **skip steps**.
	☐ I chose a **safe project** that I knew I could finish.
	☐ I am learning to **persevere** and to not give up even when tasks are challenging.
	☐ I **cooperate** well with my classmate(s)
	☐ Major setbacks or difficulties may throw me off track sometimes, but I can deal with small or medium challenges by **pausing, reflecting, and finding new solutions**.
	☐ **I use what I have learned** in one task to solve similar tasks
	☐ I did not realize how rushed I was going to be when designing possible projects.
	☐ I can **test** and **evaluate** my solutions to ensure they work
	☐ I am learning to **troubleshoot**
	☐ I **reflect** on my work and can identify my successes and failures
		Process: Strategy and Character
		
		☑ I followed the **Engineering Design Process**:
		   ☑ Brainstorm
		   ☑ Research
		   ☑ Prototype
		   ☐ Final Build (or polished prototype)
		☑ I am **willing to take risks** in creating solutions.
		☑ I **persevere** and do not give up even when tasks are challenging.
		☑ I **help** my classmate(s)
		☑ Major setbacks or difficulties may throw me off track sometimes, but I can deal with challenges by **pausing, reflecting, and finding new solutions**.
		☑ **I use what I have learned** in one task to solve new or different tasks
		☑ I take into account time and testing **constraints** when designing possible solutions.
		☑ I can **test** and **evaluate** my solutions to improve outcomes
		☑ I embrace **troubleshooting** as an opportunity for learning
		☑ I **reflect** on my work and can explain how I will improve it in the future
		☑ My reflections examine how I **am progressing as a learner**