The Lazy People: Aerial Imaging

Matyas Soukup:
Builder/ Documenter

Aryan Nagpal:
Coder/Calibration

Winson Wan:
Manager

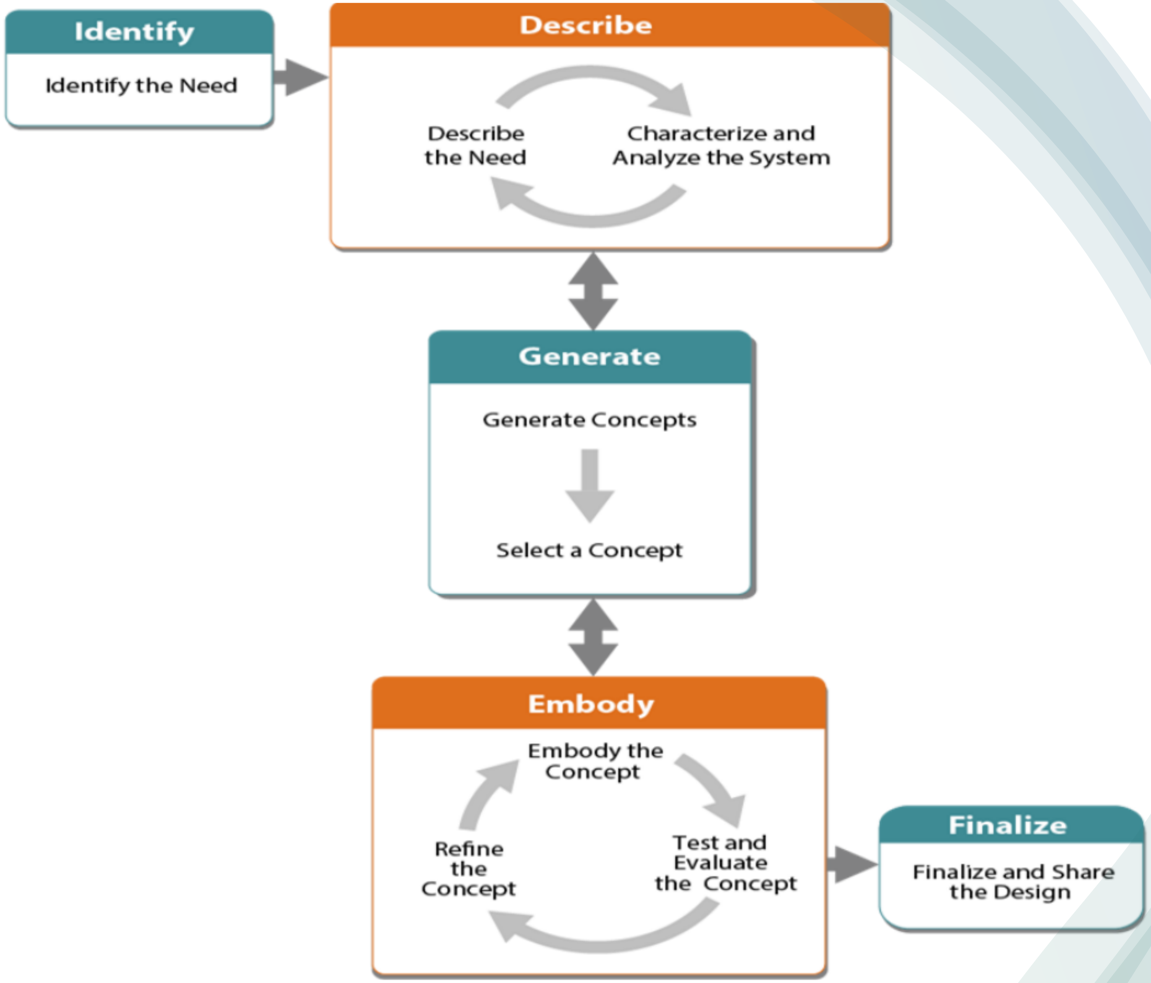
Harshveer Hooda:
Coding / Pilot

Jason Lieuallen:
Builder/Setup

Avinish Dighe:
Builder/Setup

LAZY PEOPLE

Engineering Design process



LAZY PEOPLE

Describe

Engineering Design Challenge

Your design team has been contracted to engineer an autonomous system that can be dropped to deliver emergency aid to isolated populations in the event of a disaster. Worldwide, there are over one billion people living in areas that are unreachable by roads, and delivering aid to these people requires precision and accuracy.

Your first task is to demonstrate a scaled-down prototype as a proof-of-concept. Your sample payload will be a camera that will document the descent and take photos of the landing spot.

Our Interpretation:

- System must be autonomous (no input from us after drop)
- Our prototype must be scaled down version
- Payload must contain camera to document descent
- Designing for areas that are unreachable by road
- Requires precision and accuracy

Subsystems Involved

Descent:

A system to slow down the descent of the integrated system during the drop.

Structure:

A system to contain the payload system and connect to the descent system, also to handle a light hit on the ground if needed.

Payload:

A system to signal when integrated system reaches drop altitude, landing attitude and take photos using the camera.

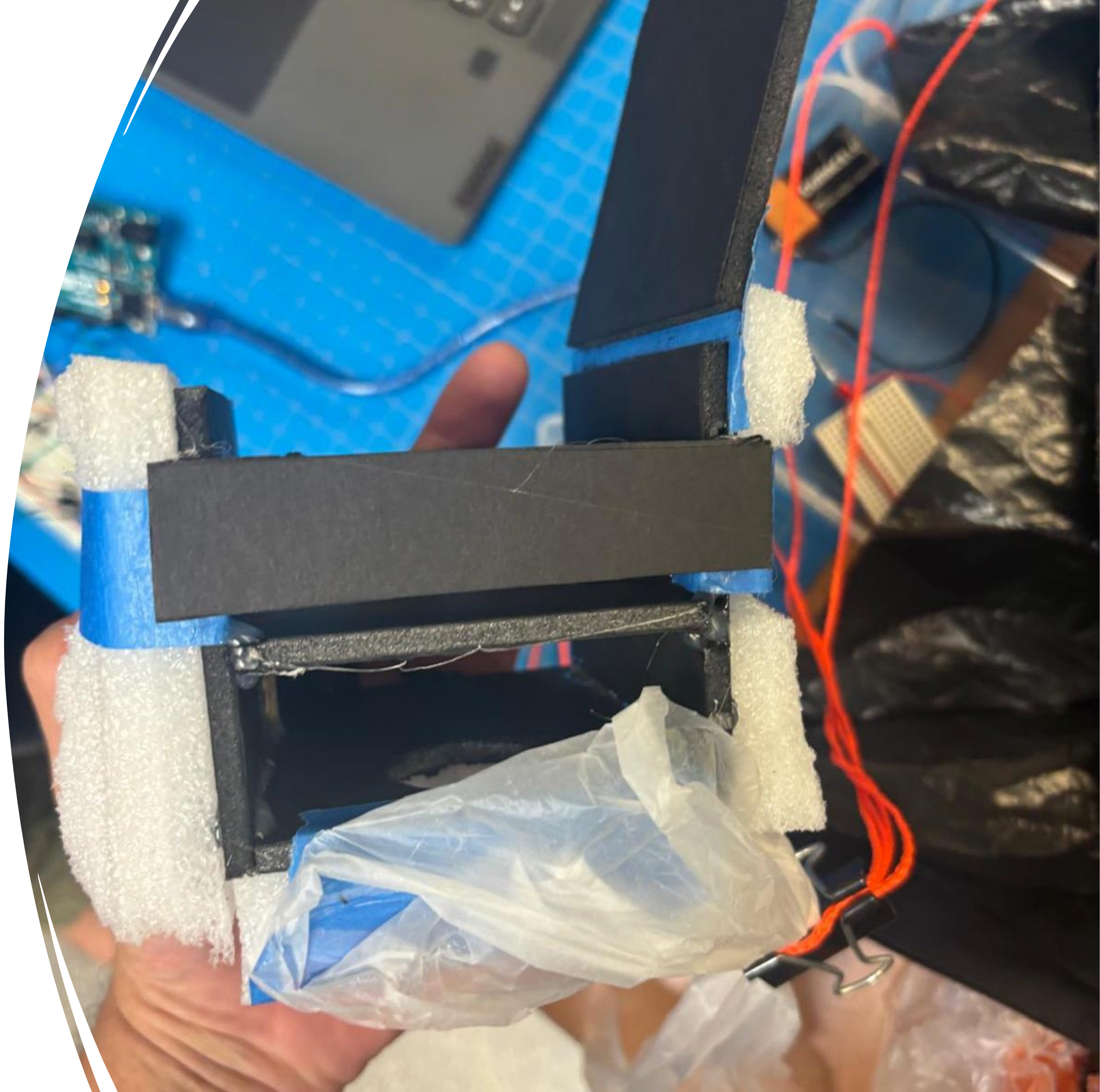
Requirements/Constraints of Integrated System

1. Be built using the metric (SI) system
2. Have a mass of no more than 350g
3. Be deployed in the same way to a predetermined altitude
4. Quickly and securely attach to the same release mechanism
5. Operate autonomously
6. Signal once it has reached target altitude
7. Start taking pictures only after it has reached the target altitude
8. Take pictures automatically
9. Control the descent
10. Maintain a velocity under 3 m/s
11. Signal once it has landed
12. Be reusable immediately after landing (without additional parts or maintenance); all components must be reusable next year in their original form.

Generate

LAZY PEOPLE

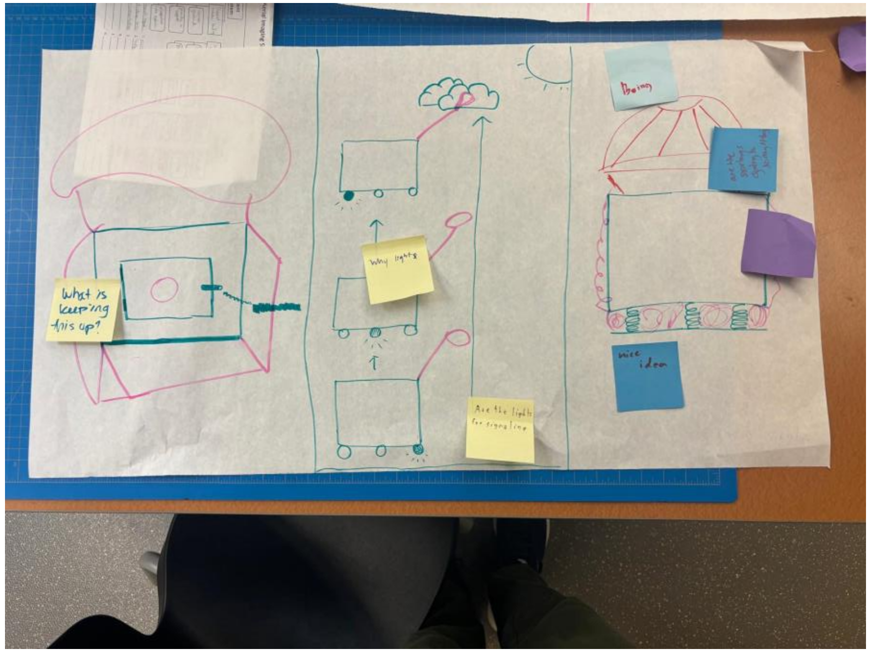
Structure

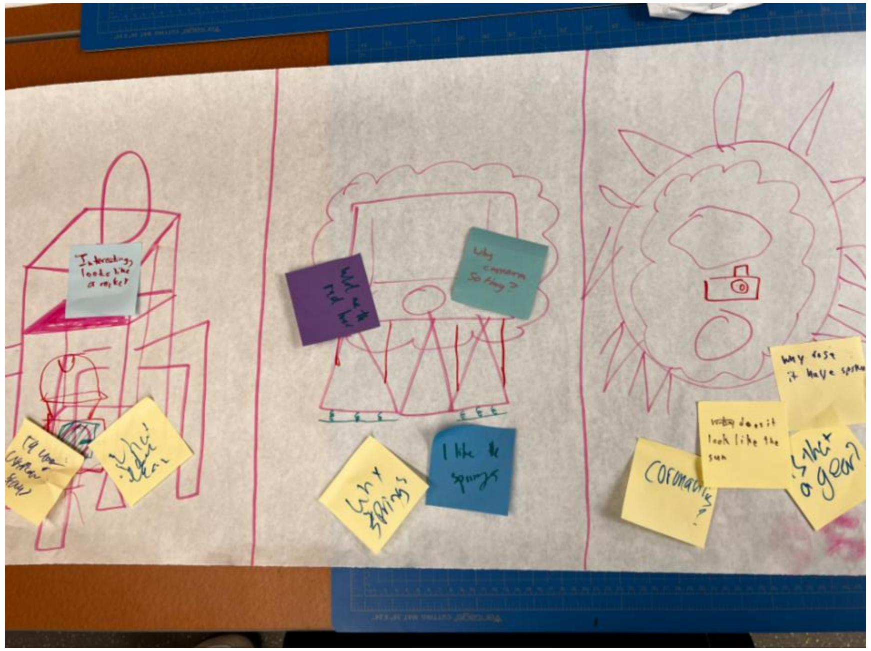


LAZY PEOPLE

Mind Map/C-Sketches

Structure Team Mind Map

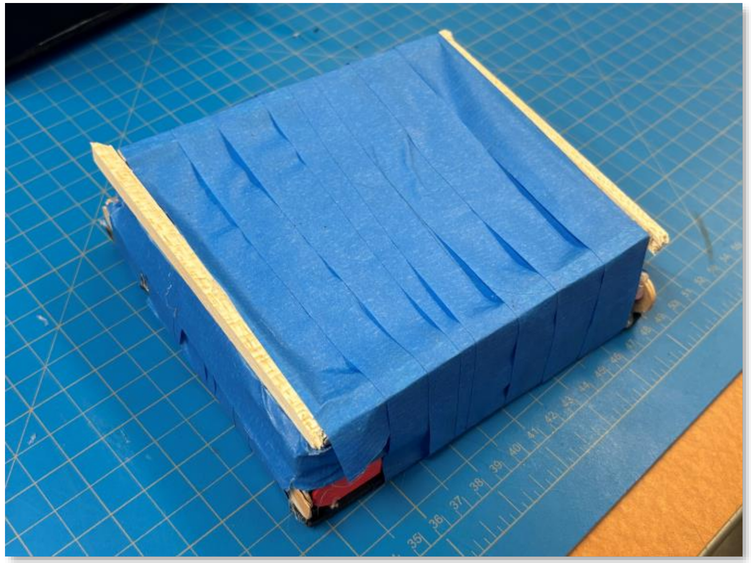


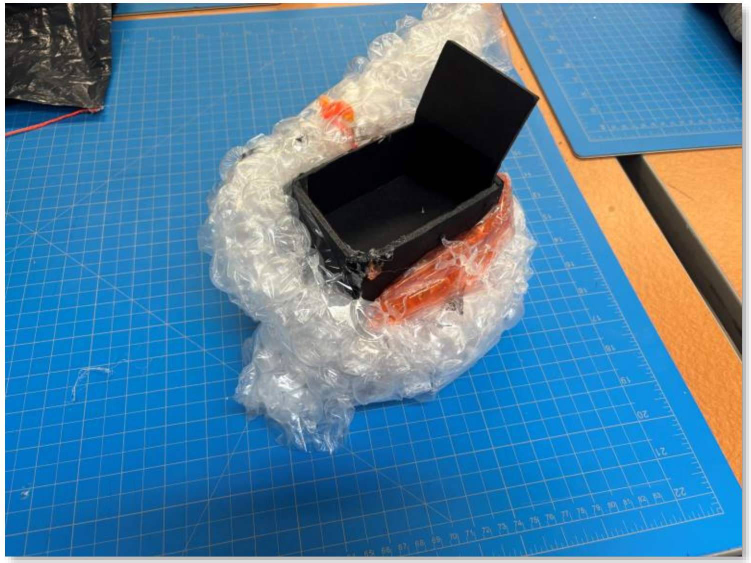


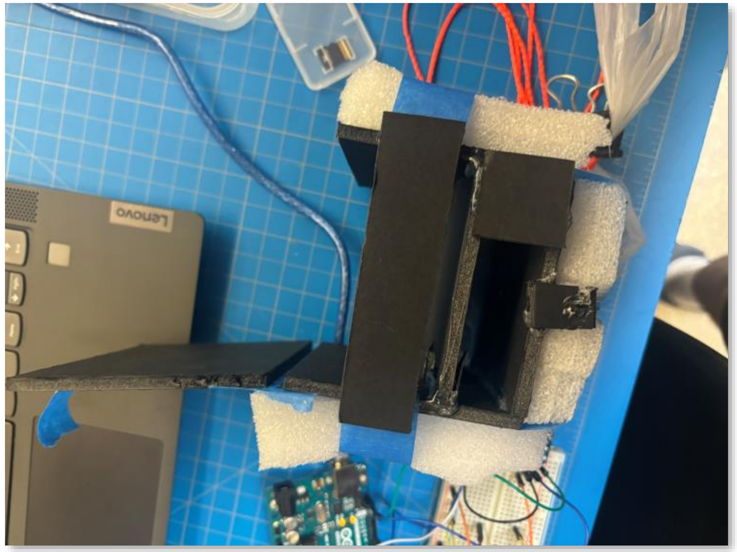
Our C-Sketches

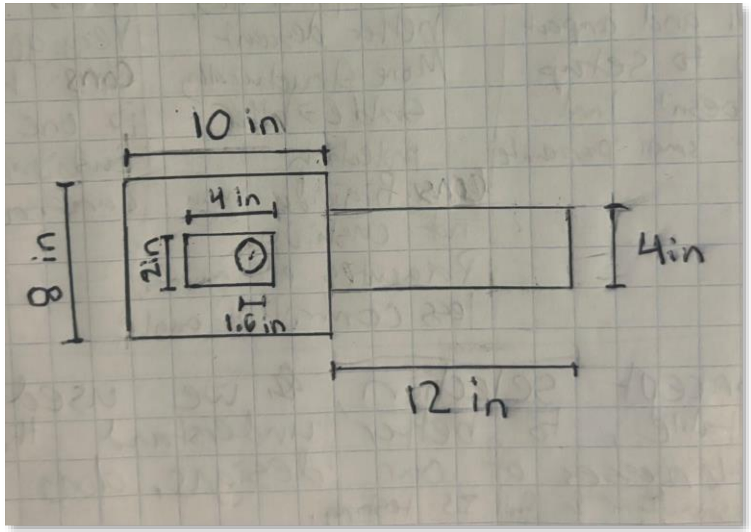
LAZY PEOPLE

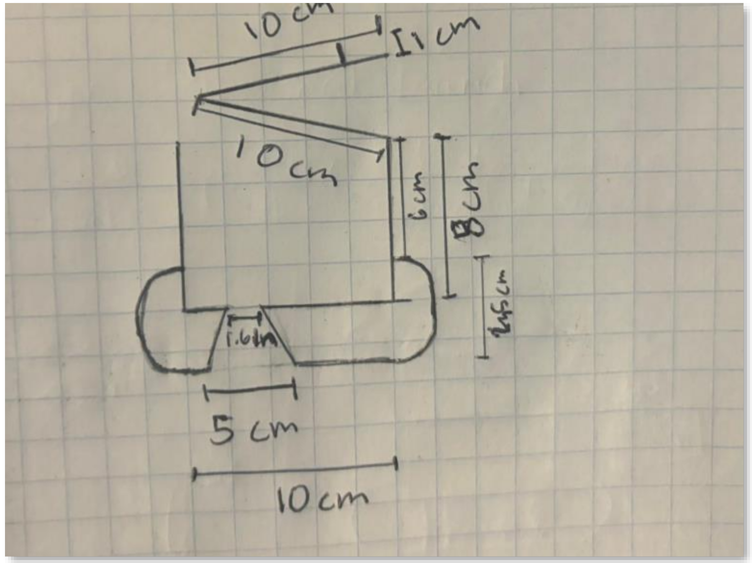
Three Prototypes

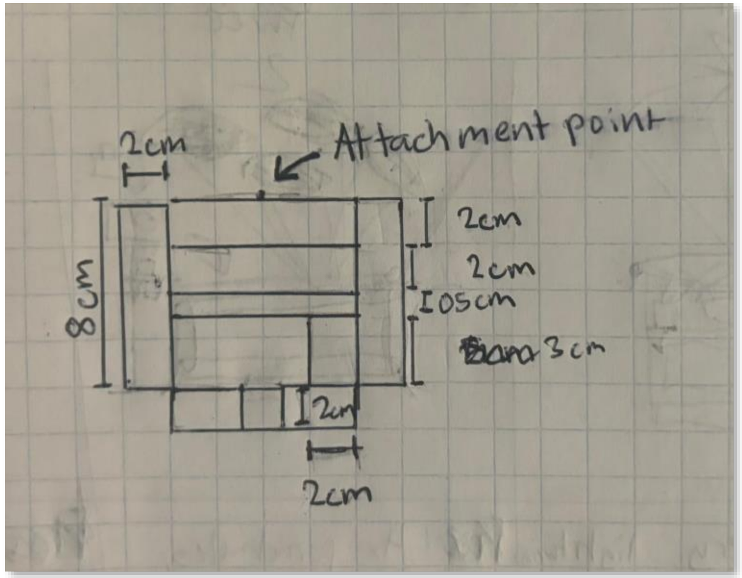








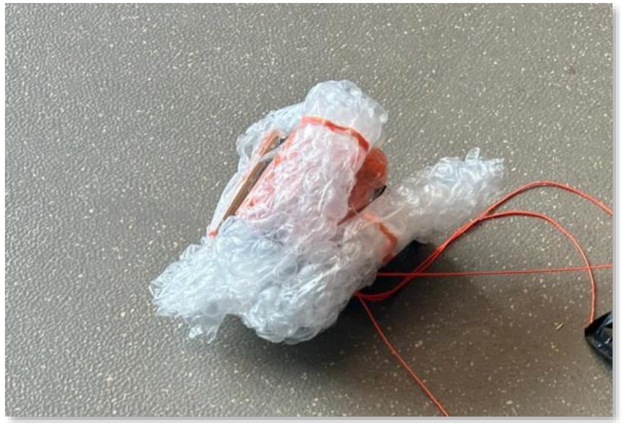




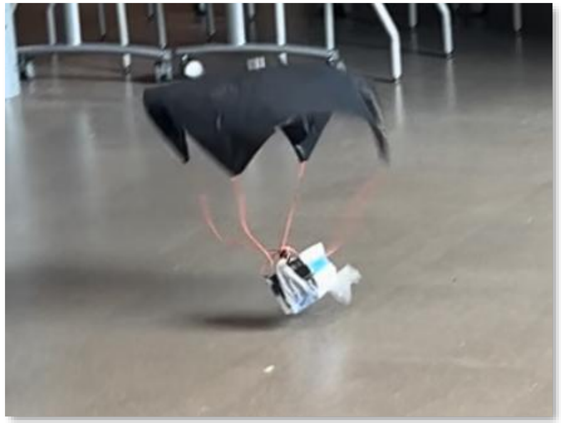
LAZY PEOPLE

Prototype testing!

- Common problem we found was that our prototypes were flipping upon landing
- We decided to have less padding on the bottom
- Prototype 3 worked really well to limit camera movement during drop, while prototype had slight shifts
- Could impact servo's ability to click button



Prototype 2 Drop Test



Prototype 3 Drop Test

LAZY PEOPLE

Data Table & Weighted decision Matrix

	Drop 1			Drop 2			Drop 3		
	Visible Damage ?	Flipped?	Soft landing?	Visible Damage?	Flipped?	Soft landing?	Visible Damage?	Flipped?	Soft landing?
Prototype 1	Yes	No	No	Yes	No	No	No	No	No
Prototype 2	No	Yes	Yes	No	Yes	Yes	Yes	No	No
Prototype 3	No	Yes	Yes	No	No	Yes	No	Yes	Yes

- This was our data tables from the drop tests we performed on our 3 prototypes
- Success ranking was loosely (best to worst) 3 - 2 - 1

This is our weighted decision matrix. We use it to decide which prototype best meets our design specs

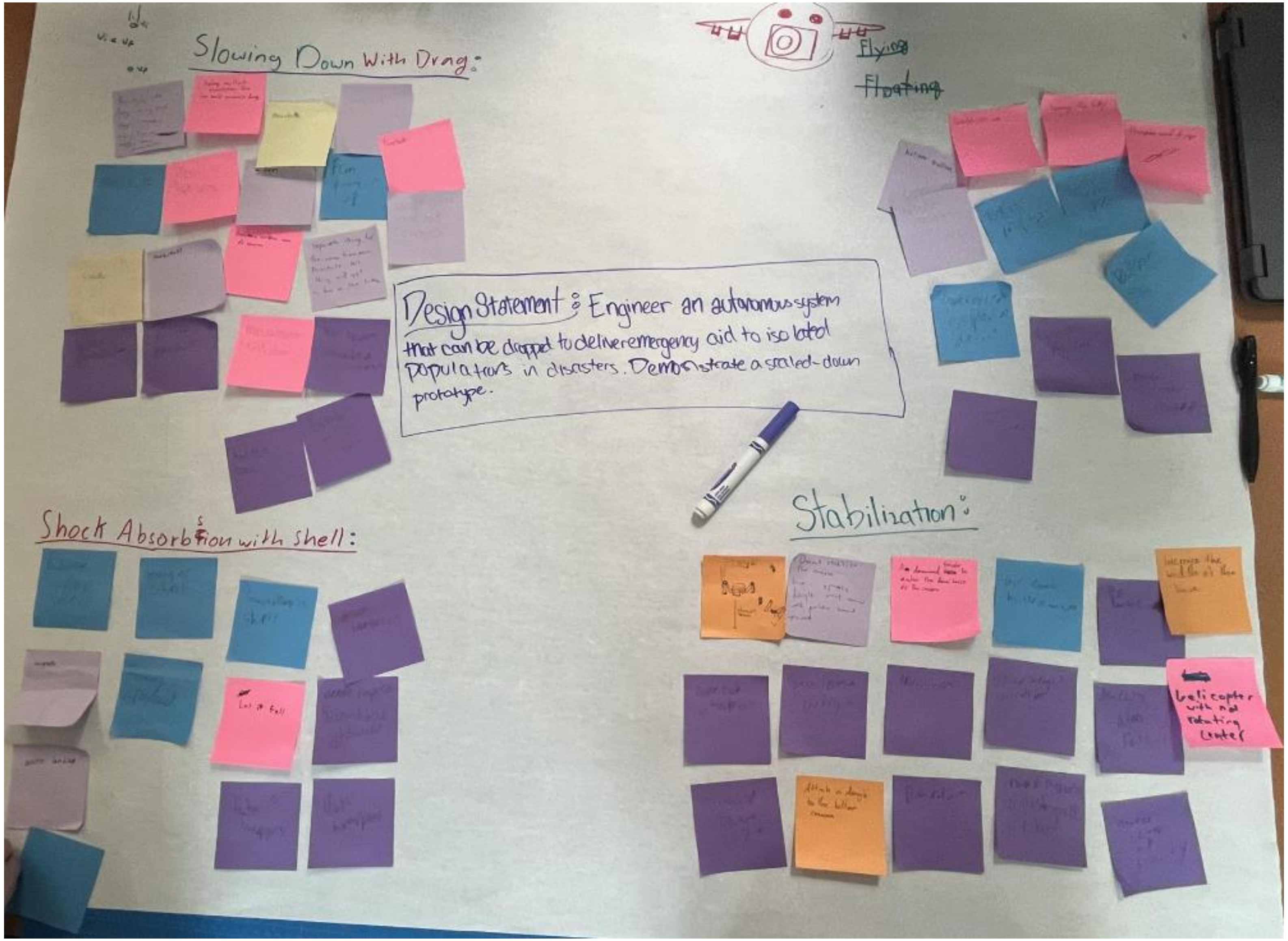
- How well it protected was the most important, so we weighted it at 40
- Compactness is important so that it fits in the bucket, so we weighted it at 10
- Lightweight designs were necessary to allow other subsystems more leeway (20)
- Reusability was a requirement, but other metrics were more important (10)
- Portability was really important for transport and setup, so we gave it (15)
- Aesthetics wasn't really a requirement, but we thought it was good to have a good-looking product

Decision Matrix	**How well it protected payload**		**How compact the design was**		**How lightweight the system is**		**How reusable it is**		**How portable it is**		**Aesthetics**		**TOTAL**	
Design	**40**		**10**		**20**		**10**		**15**		**5**		**100**	Summary and/or Rationale
Prototype 1	5	2.0	5	0.50	10	2.00	2.5	0.25	15	2.25	2.5	0.1	7.1	Prototype 1 completely exploded on impact, all contents spilled out, which means that I dont think we should use it
Prototype 2	25	10.0	6	0.60	12	2.40	9	0.90	11	1.65	2	0.1	15.7	Prototype 2 had a lot of outer cushioning, which really helped absorb the impact and make sure that the payload was protected. It had a really clever mechanism for opening, and overall worked pretty well.
Prototype 3	38	15.2	9	0.90	20	4.00	9	0.90	15	2.25	2	0.1	23.4	This was the prototype that we thought would work the best after all of our testing was complete. It protected the simulated payload really well. It was extremely compact and lightweight, but the only problem that we had with it was that it would roll.

LAZY PEOPLE

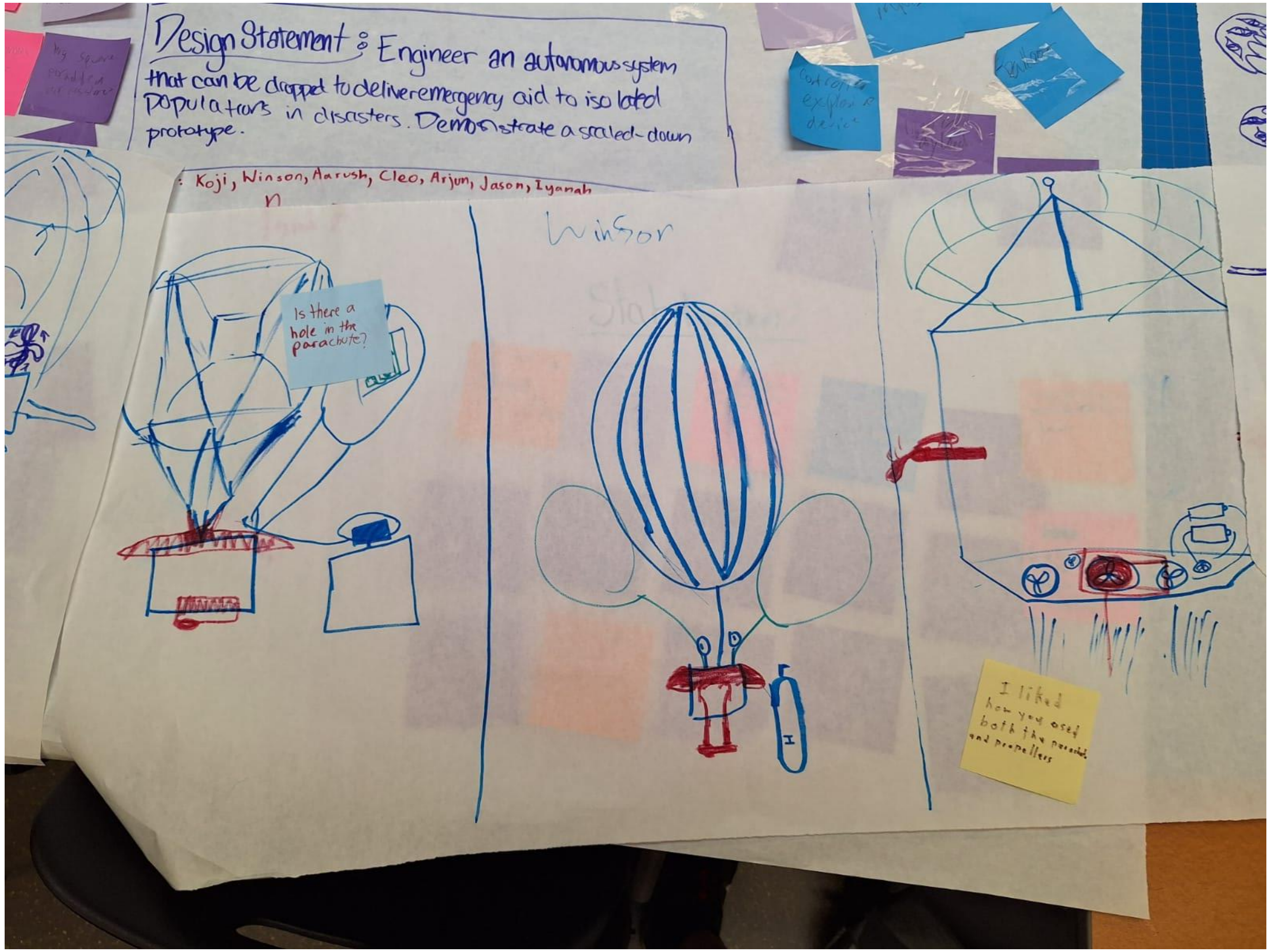
Descent

Mind Map



Design Statement: Engineer an autonomous system that can be dropped to deliver emergency aid to isolated populations in disasters. Demonstrate a scaled-down prototype.

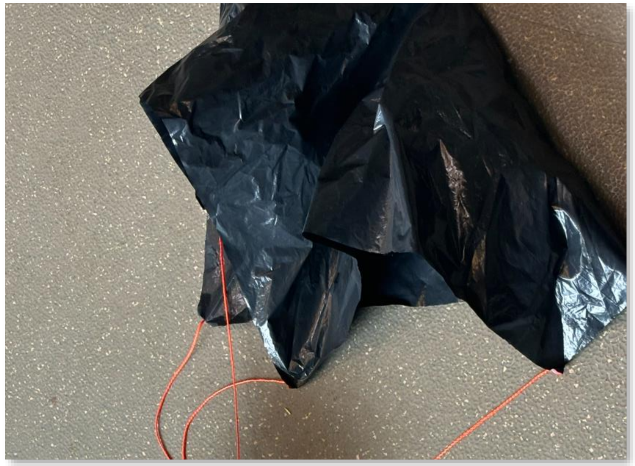
C-Sketch

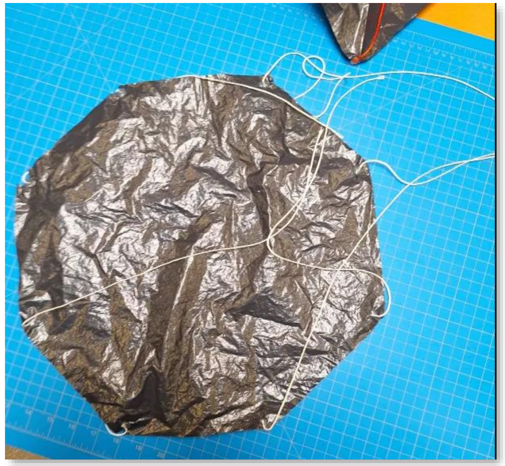


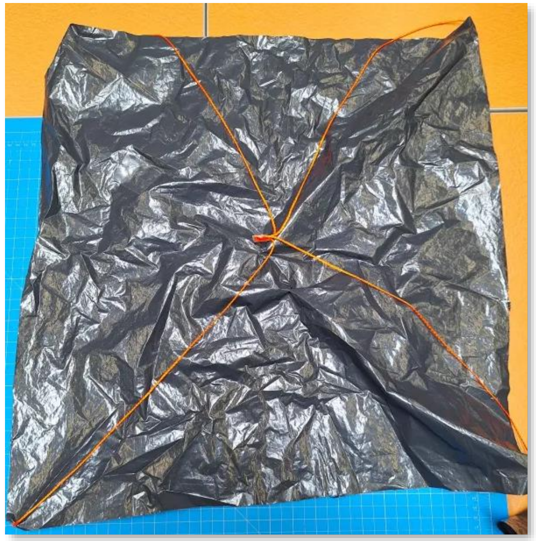
Design Statement : Engineer an autonomous system that can be dropped to deliver emergency aid to isolated populations in disasters. Demonstrate a scaled-down prototype.

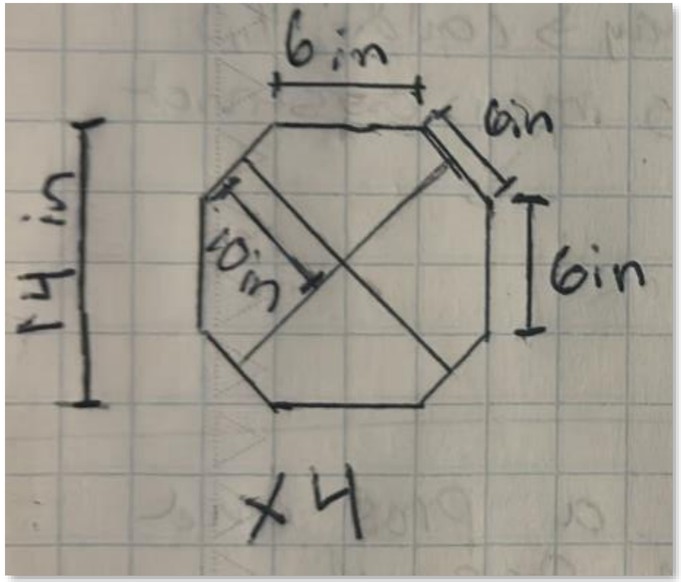
: Koji, Winson, Aarush, Cleo, Arjun, Jason, Iyanah
Is there a hole in the parachute?
Winsor
I liked how you used both the parachut and propellers

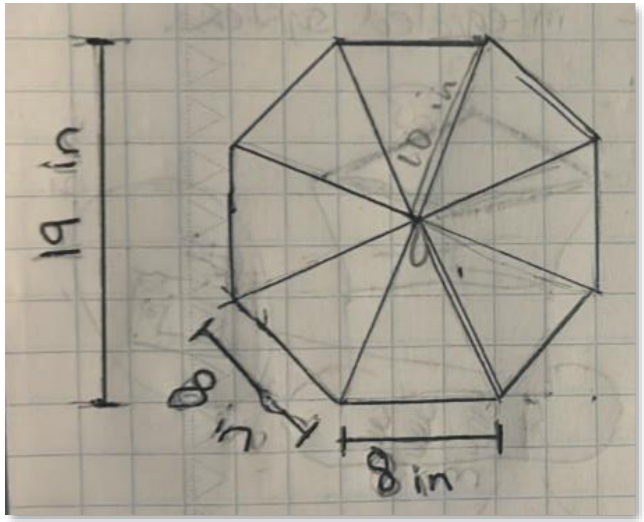
Three Prototypes

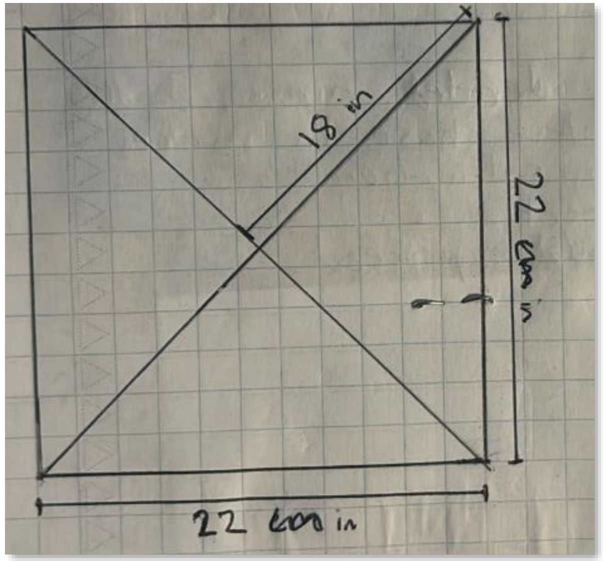












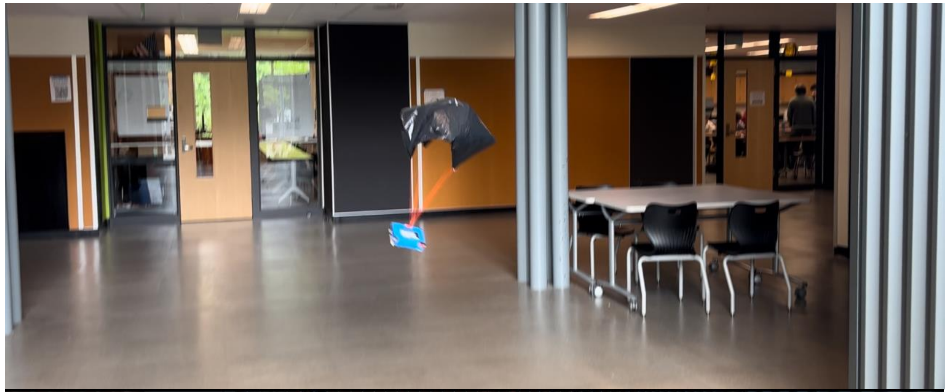
Weighted Decision Matrix

WEIGHT	35		30		10		10		10		5		100	Summary and/or Rationale
parachute	5	1.75	4	1.20	4	0.40	5	0.50	3	0.30	4	0.20	4.35	the parachute has a large suface area resulting in more air drag so it falls quite slowy but is slow to deploy and has a chance of failuer and is not very reusable
kite	3	1.05	3	0.90	3	0.30	2	0.20	5	0.50	5	0.25	3.20	a kite dosent have as much air drag drag as a parachute and isnt as stable but can be deplaoyed easily and easily reused
glider	3	1.05	3	0.90	4	0.40	2	0.20	5	0.50	5	0.25	3.30	This would not produce stable pictures which would make it so that it wouldn't work as well when taking pictures, also it isn't the best option
propeler	5	1.75	4	1.20	1	0.10	4	0.40	5	0.50	5	0.25	4.20	This is hard to build, and we would not choose this because, it is not as good as the parachute, also it is very heavy which would make it harder for the other teams to work
multi parachute	5	1.75	2	0.60	3	0.30	5	0.50	2	0.20	2	0.10	3.45	the multi parachute falls very slow but is hared to set up and could easily fail as well as deploying slowly

LAZY PEOPLE

Descent Testing

	Drop 1	Drop 2	Drop 3
Prototype 1	3.1 m/s	2.6 m/s	2.9 m/s
Prototype 2	2.6 m/s	2.8 m/s	2.3 m/s
Prototype 3	2.5 m/s	1.6 m/s	2.1 m/s

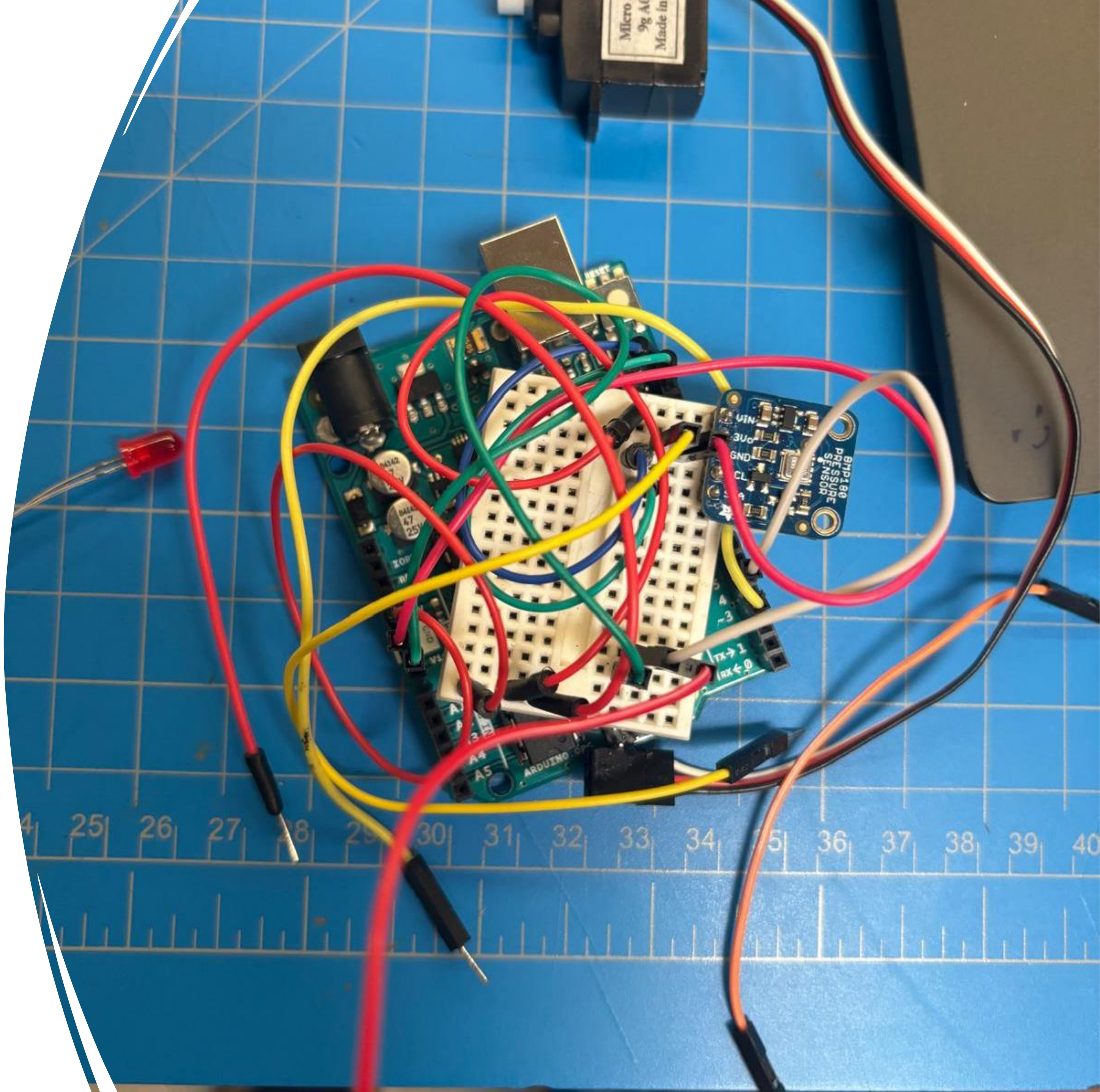


Prototype 1 went dismally, as without a full payload it was already close to the 3m/s cutoff

Prototype 2 was the one that we thought would work the best, however it didn't do as good as we were hoping

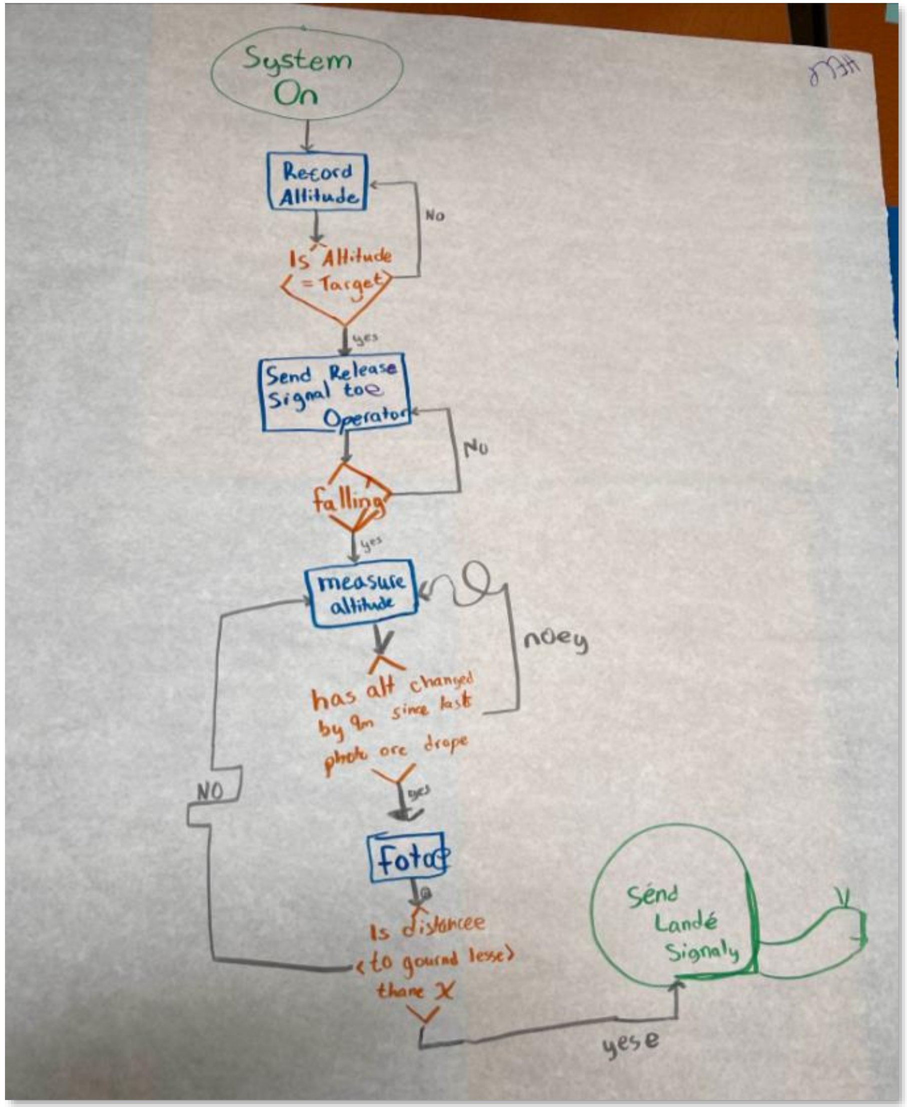
Prototype 3 performed the best, inflating really well and slowing descent down a lot

Payload


BMP180
PRESSURE
SENSOR
VIN
3Vo
GND
SCL
ARDUINO
A4
A5
TX→1
RX←0
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40

Mind Map/ Flowchart

- Helped visualize what had to be done
- Gave us better understanding of timing for certain commands that had to be sent to motors
- Allowed better code formatting, eliminated possible extraneous code that might occur without prior planning

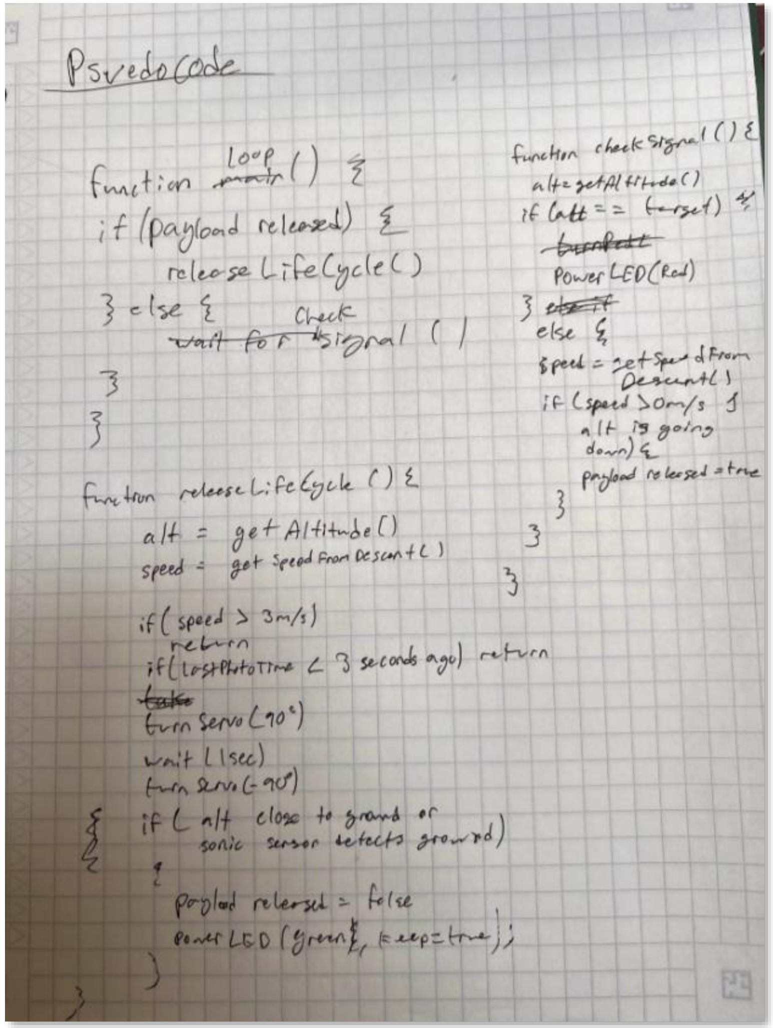

System On
Record Allitude
No
Is Altitude = Target
yes
Send Release Signal toe Operator
No
falling
yes
measure altitude
noey
has alt changed by 9m since last photo ore drope
yes
NO
Fotoe
Is distancee to gournd lesse thane X
yese
Sénd Landé Signaly

Pseudocode

While payload has not been released, constantly check if signal has been given

Once payload is released (change in altitude > 0):

- Turn servo one cycle of 60 degrees every 3 seconds (max rate of camera)
- Get altitude
- When payload has landed (change in altitude = 0), turn on “landed” LED

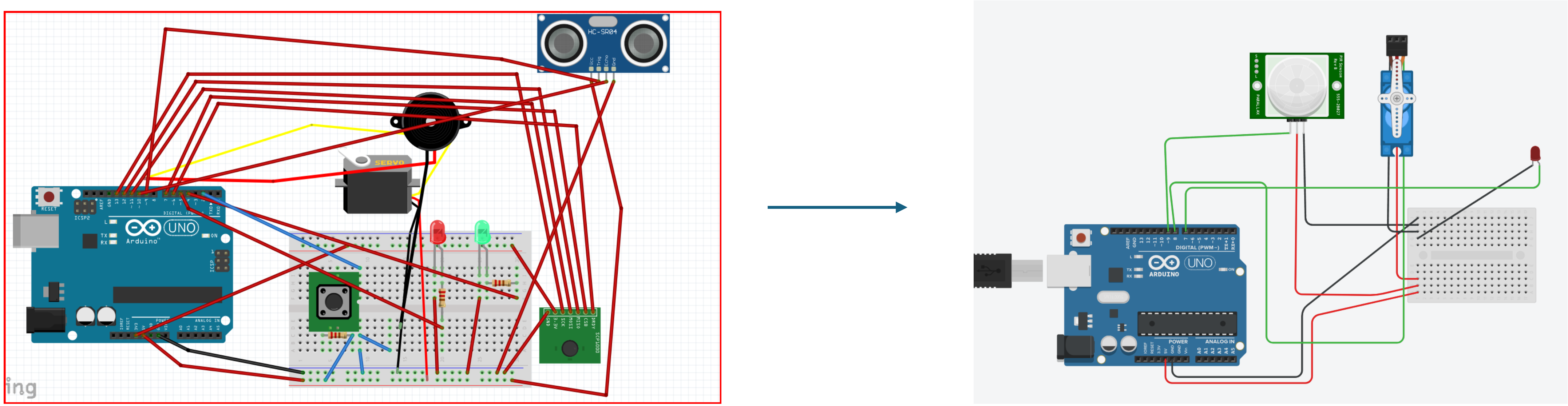


Psvedocode

```
function ~~main~~ loop() {
 if (payload released) {
     releaseLifeCycle()
 } else {
     ~~wait for~~ check signal ( )

 }
}

function releaseLifeCycle () {
    alt = getAltitude()
    speed = get Speed From Descent()

    if( speed > 3m/s)
       return
    if(lastPhotoTime < 3 seconds ago) return
    ~~take~~
    turn Servo (90°)
    wait (1sec)
    turn Servo(-90°)
{   if ( alt close to grand or
        sonic sensor detects ground)
 {
    payload released = false
    power LED (green}, keep=true});
 }
}
```

```
function checkSignal () {
  alt=getAltitude()
 if (att == target) {
    ~~burnPett~~
    Power LED(Red)
 } ~~else if~~
  else {
  speed = getSpeedFrom
              Descent()
  if (speed >0m/s {
      alt is going
      down) {
      payload released = true
   }
  }
 }
```

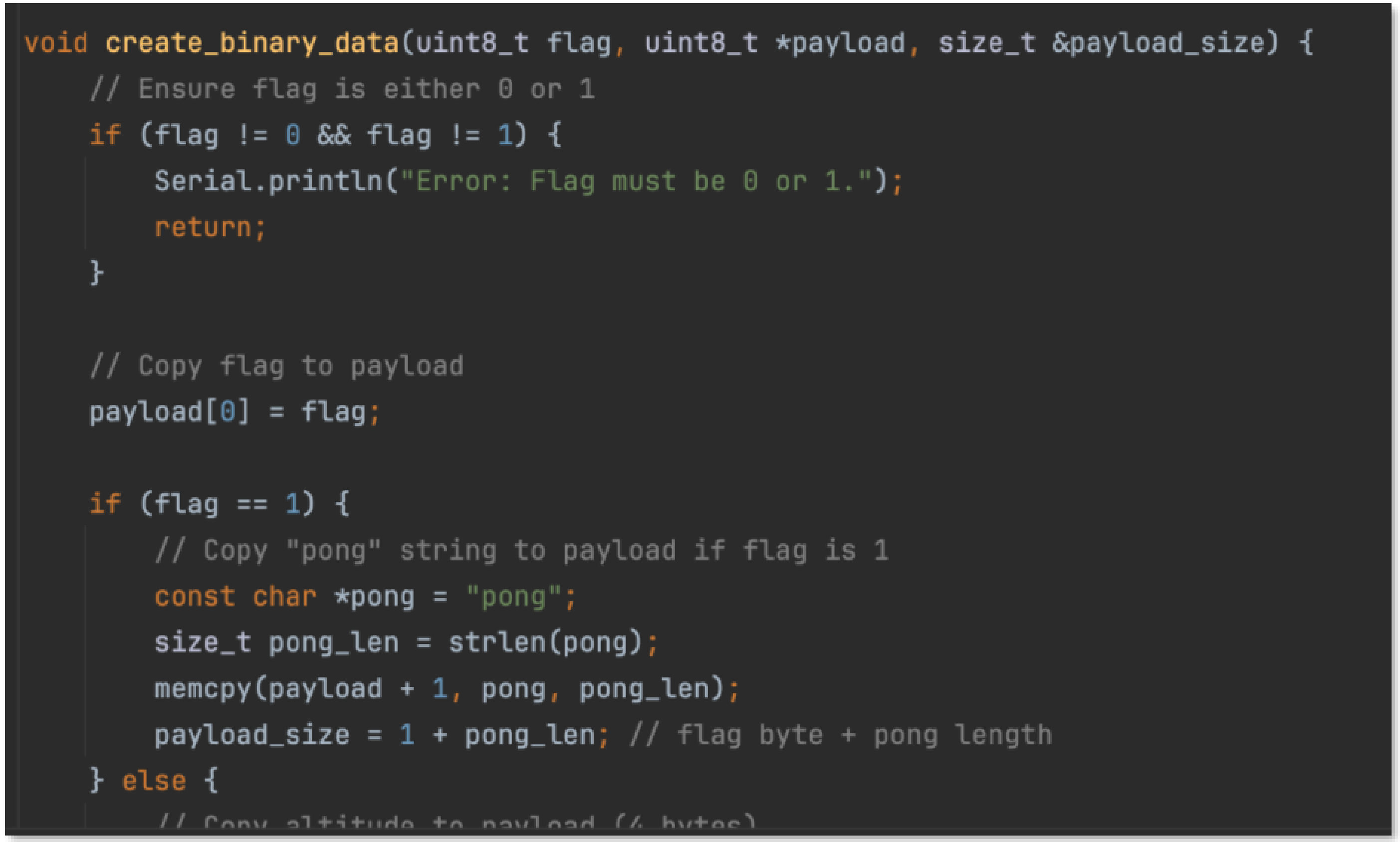
Breadboard/Arduino Working



- Final circuit had a lot of components removed that we realized we did not need
- Ultrasonic was not necessary due to alternative position tracking algorithm implemented
- Signal could be given by same LED for both max and min altitude
- Button was not necessary
- Piezo buzzer also not necessary

Code Testing/Debugging

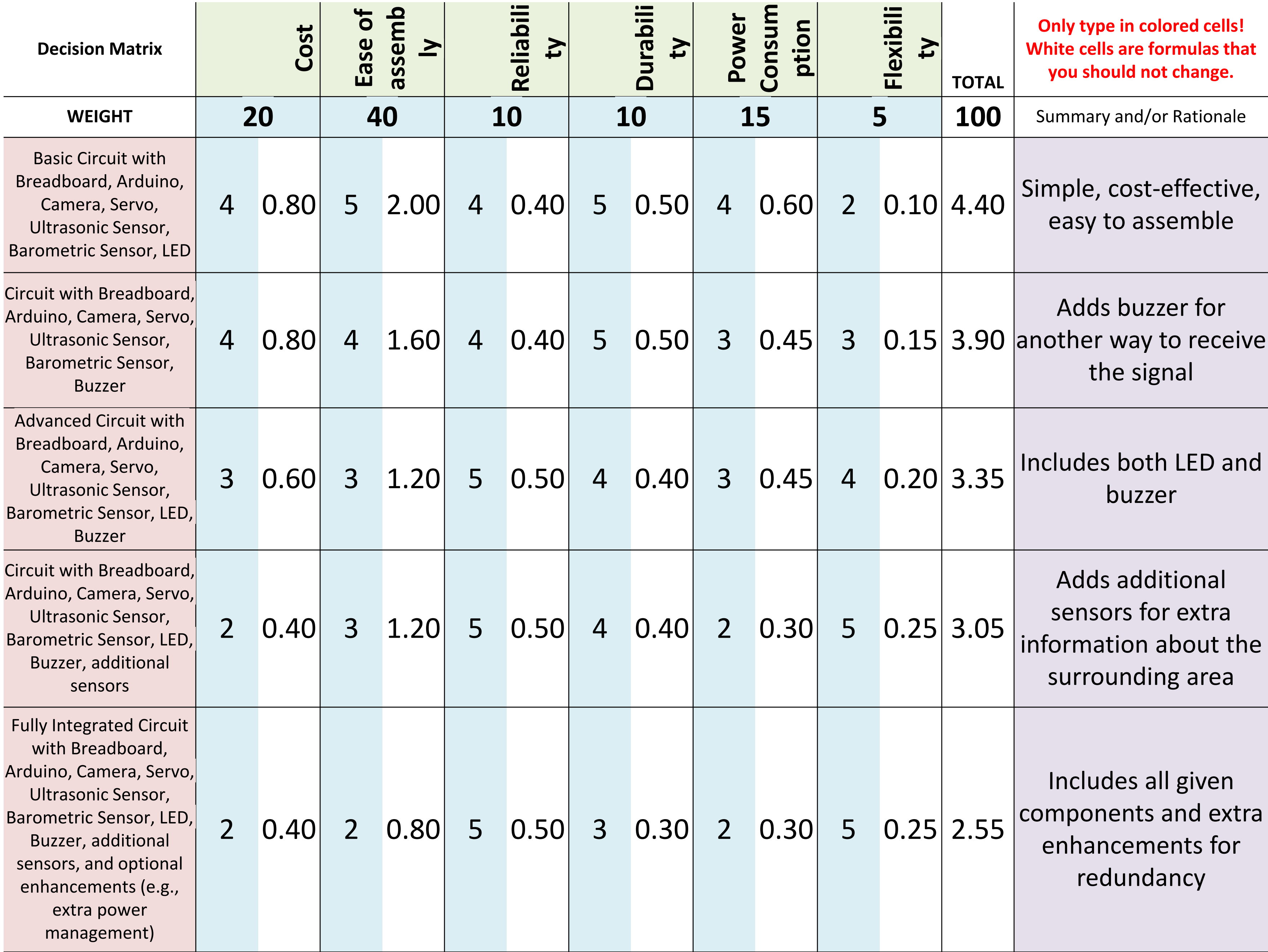
- Transceiver module communicates in binary data
- Experimented with using it as signal
- Took lots of time to figure out
- Realized it might interfere with drone waves



```
void create_binary_data(uint8_t flag, uint8_t *payload, size_t &payload_size) {
    // Ensure flag is either 0 or 1
    if (flag != 0 && flag != 1) {
        Serial.println("Error: Flag must be 0 or 1.");
        return;
    }

    // Copy flag to payload
    payload[0] = flag;

    if (flag == 1) {
        // Copy "pong" string to payload if flag is 1
        const char *pong = "pong";
        size_t pong_len = strlen(pong);
        memcpy(payload + 1, pong, pong_len);
        payload_size = 1 + pong_len; // flag byte + pong length
    } else {
        // Copy altitude to payload (4 bytes)
```

Weighted Decision Matrix

- Ease of assembly was worth 40 because we realized simplicity was key
- Power consumption was worth 15 since we only had 1 battery at the time
- Durability -> Needs to be able to withstand rocking and tumbling
- Reliability is important because it needs to work time after time

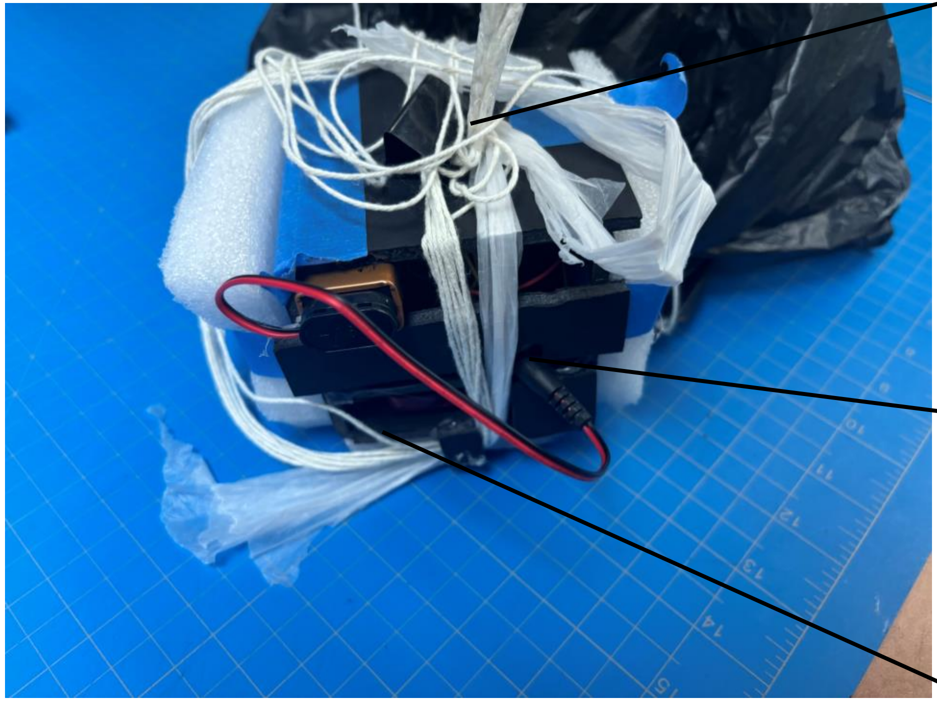


Decision Matrix	Cost		Ease of assembly		Reliability		Durability		Power Consumption		Flexibility		TOTAL	Only type in colored cells! White cells are formulas that you should not change.
WEIGHT	20		40		10		10		15		5		100	Summary and/or Rationale
Basic Circuit with Breadboard, Arduino, Camera, Servo, Ultrasonic Sensor, Barometric Sensor, LED	4	0.80	5	2.00	4	0.40	5	0.50	4	0.60	2	0.10	4.40	Simple, cost-effective, easy to assemble
Circuit with Breadboard, Arduino, Camera, Servo, Ultrasonic Sensor, Barometric Sensor, Buzzer	4	0.80	4	1.60	4	0.40	5	0.50	3	0.45	3	0.15	3.90	Adds buzzer for another way to receive the signal
Advanced Circuit with Breadboard, Arduino, Camera, Servo, Ultrasonic Sensor, Barometric Sensor, LED, Buzzer	3	0.60	3	1.20	5	0.50	4	0.40	3	0.45	4	0.20	3.35	Includes both LED and buzzer
Circuit with Breadboard, Arduino, Camera, Servo, Ultrasonic Sensor, Barometric Sensor, LED, Buzzer, additional sensors	2	0.40	3	1.20	5	0.50	4	0.40	2	0.30	5	0.25	3.05	Adds additional sensors for extra information about the surrounding area
Fully Integrated Circuit with Breadboard, Arduino, Camera, Servo, Ultrasonic Sensor, Barometric Sensor, LED, Buzzer, additional sensors, and optional enhancements (e.g., extra power management)	2	0.40	2	0.80	5	0.50	3	0.30	2	0.30	5	0.25	2.55	Includes all given components and extra enhancements for redundancy

LAZY PEOPLE

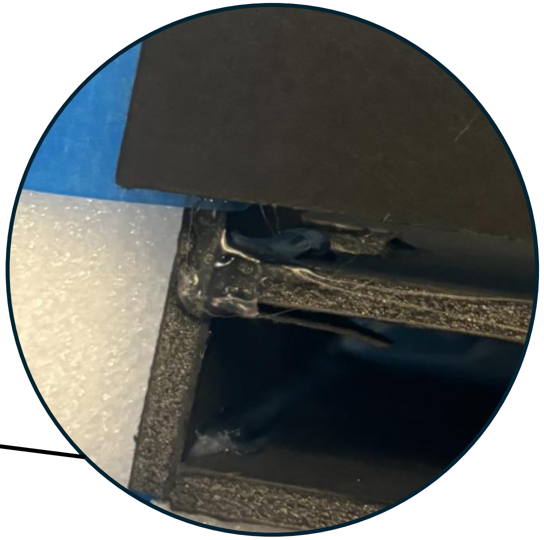
Embody

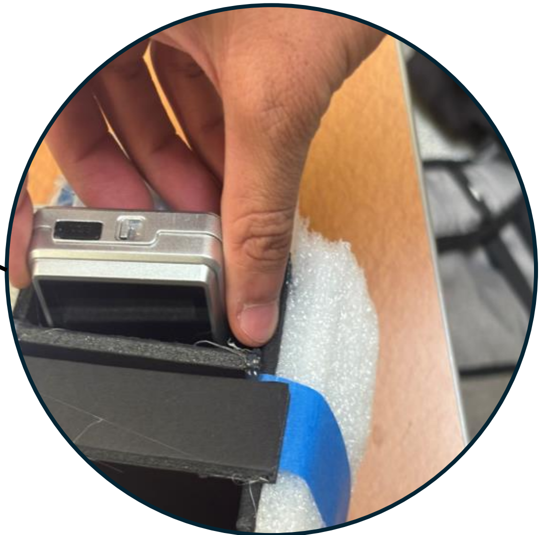
Final Integrated System



- String looped around and clipped
- Silver handles removed to save weight

-Servo housing holds it in place and restricts movement
- Single sided attachment used





- Slot for camera, easy access and secure
- Allows for free movement of camera to allow servo to push button

System Masses Table

Component	Mass (in grams)
Descent	19.6 grams
Structure	40.5 grams
Payload (Total)	207.3 grams
Camera	96 grams
Bread Board	4.5 grams
Arduino Board	30 grams
9 Volt Battery	45 grams
Altitude Sensor	4.5 gram
Servo Motor	9 grams
Red LED	1.5 grams

Total: 267.4

Con Ops Table

Step	Pilot & Setup	Payload + Final Checks	Payload Sensor Calibration	Manager + Release Manager	Set Up	Documenter
Prelaunch	Harsh	Aryan	Avnish	Winson	Jason	Matyas
1	Grab controller	Get transceiver module	Bring laptop with code	Bring extra clips for parachute	Grab Stand phones	Bring Main Phone + Back up phones
2	Calibrate the sensors	calibrate transceiver module	Evaluate Camera functionality	Evaluate release mechanism, Bluetooth device connection	Grab Descent Subsystem	Setup back up camera recorder on phone stand
3	Check off that payload + final checks person has what they needed to grab	Check off with payload and sensor calibration has what they needed to grab	Check of with manager + release manager has what they needed to grab	Check of with set up has what they needed to grab	Check off with documenter that they have what they need to grab	Check of with pilot and set up that they have what they need to grab
4	Figure out wind and sun conditions, and communicate it to the documenter	Check of with everyone in the group that they have what they need to grab.	Ensure all materials are present	Evaluate release mechanism	Help set up Video equipment	Set up Video equipment (Phone on phone stand) and main phone
5	Ensure camera is turned on	Ensure servo motor is pushing the right way	Ensure all sensors are responding	Make sure descent systems strings are untangled.	Ensure camera angle is correct angle	Documents surroundings
6	Ensure that the little piece of foam making sure the camera does not move is in place.	Ensure latest code has been downloaded	Turn on & off the Arduino once	Make sure there are no holes in the descent system.	Ensure that recording has started	Take photos of launch setup
7	Ensure the plastic wrapping keeping structure closed is tightly wrapped and looped closed	Ensure all wires are connected	Ensure wires are taped down to sides bottom.	Check in with Structure system to make sure it is sealed and ready.	Survey site for known hazards Record mental observations of setup / what could have	Evaluate the photo camera to ensure it is in the right mode and ensure that recording has started.
8	Ensure that everything is ready to be stacked properly	Make sure everyone is ready to start the putting the integrated system into the bucket	Tell Mr. Bonomo that we are ready to start the launch process (Putting the things into bucket for launch)	Give final heads up that everyone’s job is done	Check weather on phone to ensure no rain	Set camera focus appropriately
9	Put structure + payload into bucket.	Close release mechanism once string is attached.	Ensure no movement of sensors	Attach string to release mechanisms	Ensure that pilot has certificate permit	Take multiple pre-launch shots of setup

LAZY PEOPLE

Con Ops Table

Launch						
1	Communicate with release manager to make sure he is ready	Ensure stability of all connective parts	Check to make sure everyone is ready for a launch	Ensure flight plan has been made	Ensure no obvious structure failures	Take pictures the whole time
2	Start Countdown	Survey box for any dead giveaway hazards	Make sure the release manager has the release mechanisms button	Make sure hand is ready on the release button	Check To make sure the camera is on	Start recording on phone on the phone stand
3	Use drone to life out of bucket	Check To make sure everyone is in their right spot	Keep everyone away from the drone area	Make sure everyone is ready for their job	Make sure everything is ready	Start recording on main phone
4	Fly drone up	Keep everyone away from the drone area	Look up for the signal as a backup	Grab bucket away	Take Photos of everything	Tell payload and sensor calibration what to write down for observations
5	Fly the drone, communicate with harsh to know altitude	Wait for connection	Call out live altitude readings and help documenter write down observations	Wait for signal to release the Integrated system	Check to make sure cameras taking video of the process is still on and working	Tell payload and sensor calibration what to write down for observations
	Keep drone stable	Ensure no disruption in connection	Call out live altitude readings and help documenter write down observations	Wait for signal to release the Integrated system	Watch for any visible structure failure	Tell payload and sensor calibration what to write down for observations
6	Adjust to meet flight plan	Verify Altitude callouts	Call out live altitude readings and help documenter write down observations	Wait for signal to release the Integrated system	Callout any structure failure	Tell payload and sensor calibration what to write down for observations
7	Get ready to drop	Callout any discrepancies between drone and sensor altitudes	Call out live altitude readings and help documenter write down observations	Wait for signal to release the Integrated system	Ensure flight plan is going correctly	Tell payload and sensor calibration what to write down for observations
8	Wait for drop callout	Ensure stability between sensor and drone	Call out live altitude readings and help documenter write down observations	Wait for signal to release the Integrated system	Callout flight plan discrepancies	Tell payload and sensor calibration what to write down for observations
9	Final position check	Ensure components are working (briefly)	Call out live altitude readings and6help documenter write down observations	Wait for signal to release the Integrated system	Countdown for drop	Tell payload and sensor calibration what to write down for observations
10	Confirm drop	Wait for drop	Call out live altitude readings and help documenter write down observations	Wait for signal to release the Integrated system	Wait for drop	Tell payload and sensor calibration what to write down for observations
11	Wait for touchdown before descent	Watch altitude decrease speed	Call out live altitude readings and help documenter write down observations	Wait for signal to release the Integrated system	Record visual observations on structure specifically	Tell payload and sensor calibration what to write down for observations

LAZY PEOPLE

Con Ops Table

Post-launch						
1	Pilot + Set up	Payload + Final checks	Payload + Sensor calibration	Manager + Release manager	Set up	Document
2	Wait until complete touchdown	Check for general damage of Arduino components	Check Lid and Strap	Check for holes in the descent system	Check Recording of the backup phone	Take picture of final landing
3	Turn off drone propellers	Check for general damage of Arduino components	Ensure no structural damage	Check the strings on the descent system	Help documenter with notes.	Record flight observations
4	Approach drone after propellers have completely stopped spinning.	Evaluate each Arduino part for function note down condition they are in and how well they work	Ensure that latch part is still intact	Check for damage on the camera	Meet with documenter for notes	Record the weather conditions end of launch
5	Wait for recovery checklist run through	Evaluate each Arduino part for function note down condition they are in and how well they work	Ensure no changes to camera snug fit	Check for the photos on the camera	Share notes on from documenter to everyone	Record any environmental changes
6	Decide if another launch is doing	Evaluate each Arduino part for function note down condition they are in and how well they work	Ensure that all foam is not broken.	Make a list of everything broken and working and how well and decide if going for another launch do we change descent system.	Help change the descent system if needed.	Talk to pilot as recover checklist is being run through to record flight experience
7	Work with Release tech to rebook the parachute	Evaluate each Arduino part for function note down condition they are in and how well they work	Make a list of everything working well and everything. broken	Analyze the list	Help change the structure if needed	Talk to
8	Start group on Loop back around to prelaunch if another launch is chosen.	Give notes on each Arduino part to Manager (Winson)	Give list to manager (Winson)	Analyze the list to decide next step on what to do about the Payload and payload	Help change the payload	Document needed information

LAZY PEOPLE

Integrated system: FMEA table

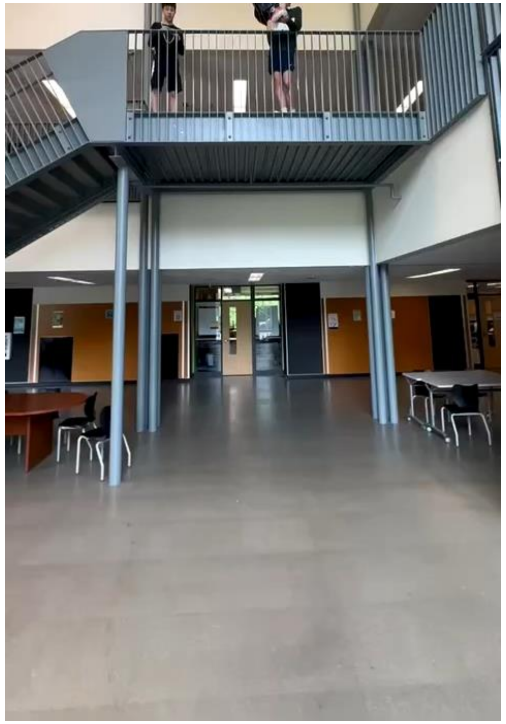
	Item/Function	Potential Failure Mode	Potential Causes of the Failure	Potential Effects of the Failure	Severity of Occurrence	Probability of Occurrence	Recommended Actions
Questions	***What is the component, and what are its functions and requirements?*** *List all for the subsystem.*	***What can go wrong?*** *- no function* *-partially degraded function* *-intermittent function* *-unintended function*	***What would cause the failure?***	***What could happen if the failure occurs?***	***How severe is the failure?*** *5: catastrophic; mission failure* *1: some damage but mission proceeds as planned*	***How likely is the failure?*** *5: near certain (80-100%)* *4: highly like (60-80%)* *3: likely (40-60%)* *2: unlikely (20-40%)* *1: improbable (0-20%)*	***What can be done to prevent the failure from occurring?*** *- Design changes* *- Operational changes* *- Communication changes*
System level	Parachute Mount comes undone	partially degraded function	The clip loosing tension	Box opens	5	3	Design change
	parachute string tangle	partially degraded function	Rocking of the entire system during descent	System will fall way too fast	4	2	Design change
	Parachute does not open	No function	Parachute folds during launch or gets caught weird on wind	Entire system plummets in freefall	5	1	Operational change
	Servo gets stuck	Partially degraded function	unintended in flight movement	Camera does not take pictures	5	3	Communication change
	servo doesn't turn enough	Unintended function	unintended in flight movement / uncalibrated sensors	No pictures	5	4	Communication change
	Wires come undone	No function	Wobble during descent or ascent	Sensors / servo do not function, no signals are provided	3	5	Design change
	Box Strap breaks	partially degraded function	Latch comes undone	box tilts and can open, causing contents to fall out	5	3	Design/operational change
	Box snaps/cracks/breaks	Partially degraded function	Bubble wrap gets popped/hits something its not supposed to	Contents could fall out, or wouldn't have adequate protection	3	4	Design change
	Camera moves out of place	Partially degraded function	Rocking of the entire system during descent/ unintended in flight movement	may misalign camera and jam servo or other components	2	3	Design change
	Signal is not conveyed	No function	Code issue or sensor not working	Would not know when to release mechanism	3	5	Operational change/communication change

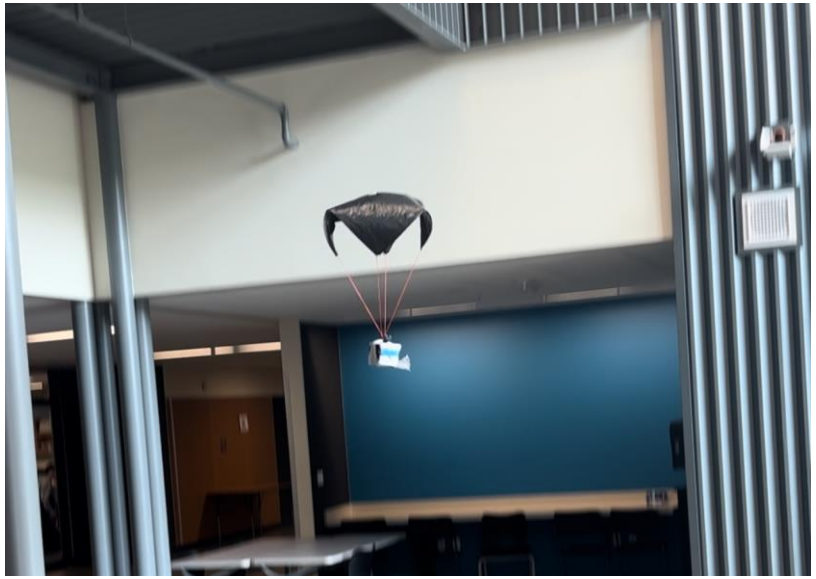
LAZY PEOPLE

Prototype testing: Balcony launches

These are two out of three of the balcony launches we did as testing

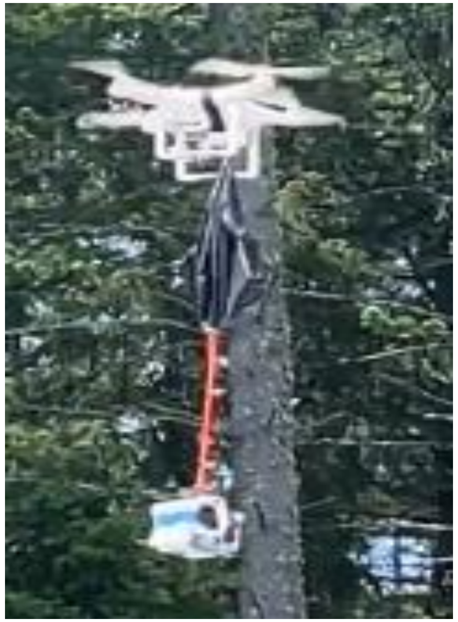
- Structure prototype 1 exploded
- Structure prototypes 2 & 3 took a beating, but protected their simulated payload well
- Descent was really fast -> Need to fix
- Need to fix rocking problem, could impair picture taking capabilities





LAZY PEOPLE

Prototype testing: First Launch day



Prototype testing: First Launch Retrospective

Issues:

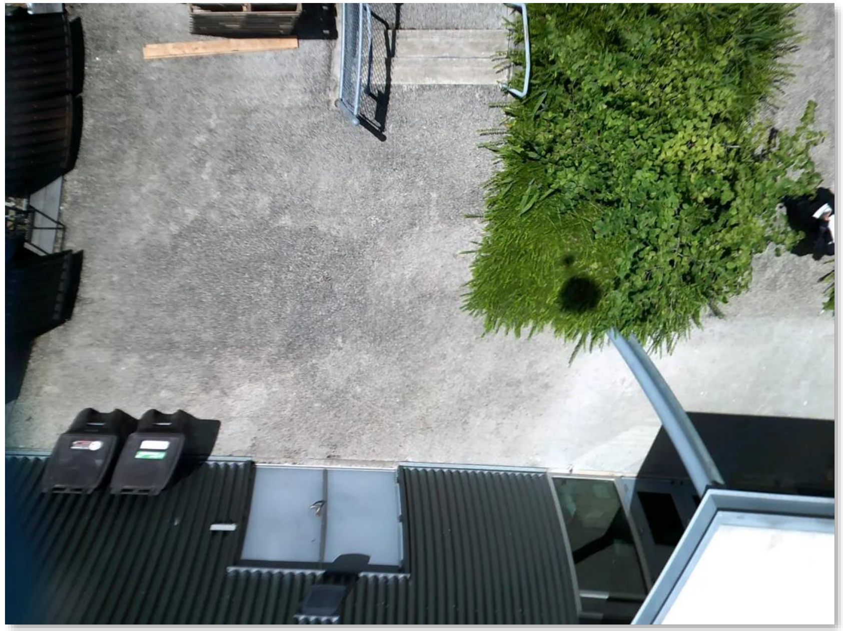
- Realized that we need to improve communication
- Some mishap with knowing roles
- Missed some crucial events like removing bucket and descent altitude callouts
- Camera did not take pictures
- Servo was not turning enough -> Camera being clicked but not hard enough
- First launch had a speed of 3.5 m/s for descent, which was way too high
- Landing was very sudden and rough
- System flipped on impact
- Landing signal did not turn on

Good things:

- Weight was comfortably under
- Altitude reached signal worked well
- Parachute held camera stable during descent
- Box did not spin during ascent

Finalize

Final Launch Pictures



LAZY PEOPLE

Overall Retrospective

- After our first launch we started to understand our roles better
- Overall record pit crew time of 2.8 seconds.
- Descent speed down to 2.9 meters per second,
- best of all we got the camera to work.
- Everything started to fall into place after the first launch and by the final launch everyone understood what needed to be done and did it.

Individual Challenges

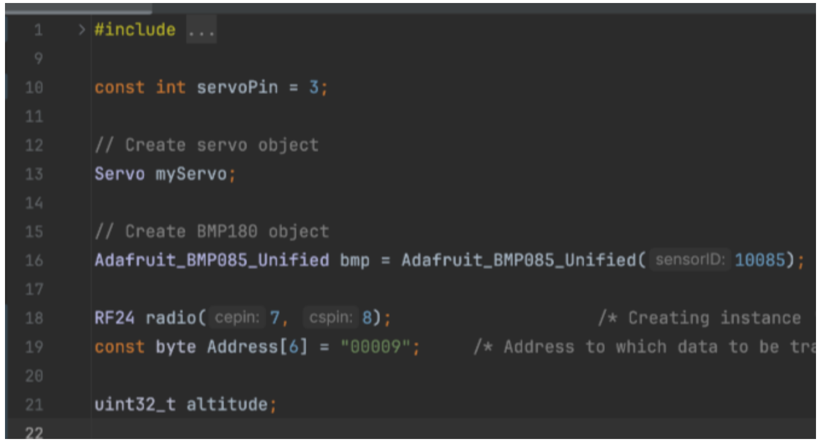
Structure

Finding a way for it to be integrated with the descent mechanism

Descent

Best working parachute got punctured and got a hole in it

Payload



```
#include ...

const int servoPin = 3;

// Create servo object
Servo myServo;

// Create BMP180 object
Adafruit_BMP085_Unified bmp = Adafruit_BMP085_Unified( sensorID: 10085);

RF24 radio( cepin: 7,  cspin: 8);                    /* Creating instance
const byte Address[6] = "00009";      /* Address to which data to be tra

uint32_t altitude;
```

Servo motor was getting stuck while trying to push the camera button

Thank you for listening to our presentation!