ECE 445

Senior Design Laboratory

Final Report

Vertical Climbing Drone

Team No. 40
Jacob Corsaw
(jcorsaw2@illinois.edu)

Jeffrey Chang
(jdchang3@illinois.edu)

TA: Kai Chieh (Jeff) Chang

December 6, 2023

Abstract

This is the final report for Team 40's Vertical Climbing Drone project for the fall semester of ECE 445. In this report, we provide an overview and details about the design, implementation, problems, solutions, and cost of our project.

Contents

1 Introduction

In this section we aim to provide context to the problem and how our project provides a solution.

1.1 Problem

For about the past decade, drones have become more available and more widely used in many commercial, industrial, and domestic applications. These drones have allowed us to see and examine situations that a human could not with unprecedented freedom. Specifically, we can now use drones to scope out crawl spaces, vents, pipes, and other tight environments where it is both hazardous and would require much more work to have a human inspector. The commercial and industrial sectors have already adopted this new capability, where we can see that "pipe climbing robots have multiple usages such as inspection in chemical industries or for public sewers" [1] among other, more domestic uses. That being said, these drones are nearly all of a similar build: wheels or tracks to crawl along the floor. However, in vents and pipes, we put bends in them to change elevation. A tracked or wheeled drone that rides along the floor will be unable to move any further, as it would get stuck on the upward bend or be lost if it went downward through a vertical shaft.

1.2 Solution

Our solution aims to change this. We created a similar tracked drone that is a fraction of the size of most commercial drones, and also has a top-mounted arm that moves in a radius that serves to expand the size of our drone. This swinging top arm is able to move in such a way that it can push against the "ceiling" of a confined space so that it will always be able to provide a downforce relative to the tracks. The arm keeps our drone in contact with whatever the tracks are attempting to climb, giving it the ability to climb a vertical shaft. Our modifications to the standard tracked drone have given it an unprecedented degree of mobility, and turns the regular 2-dimensional traversal tool into a 3-dimensional explorer. This makes our project a prime traversal tool to scout ventilation ducts, pipes, and wall spaces for inspection. In addition, we've added the ability for our drone to be used as a wire-guiding device, so that users can attach cables or wires and have the drone feed them through areas that are impossible to just "feel out" with a line. Therefore, our project would be right at home to be the ideal tool for any job that requires work within a confined space.

1.3 Visual Aid

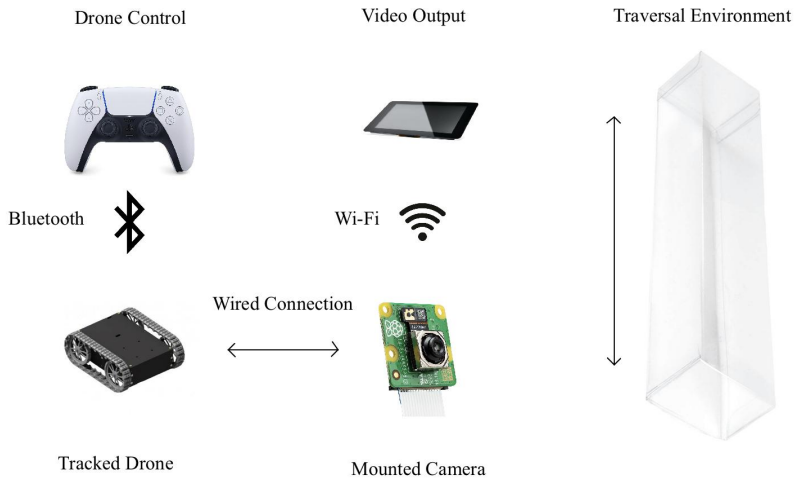


Figure 1: Visual Overview of Project

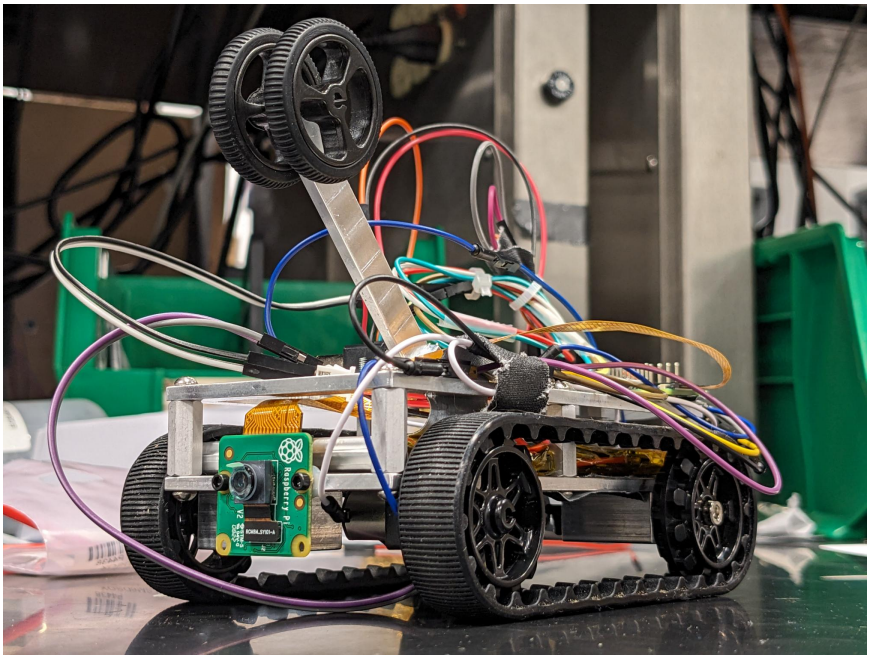


Figure 2: Final Result

-This is the final result of our project. The camera is featured on the front, along with the two tracks and the upper arm.

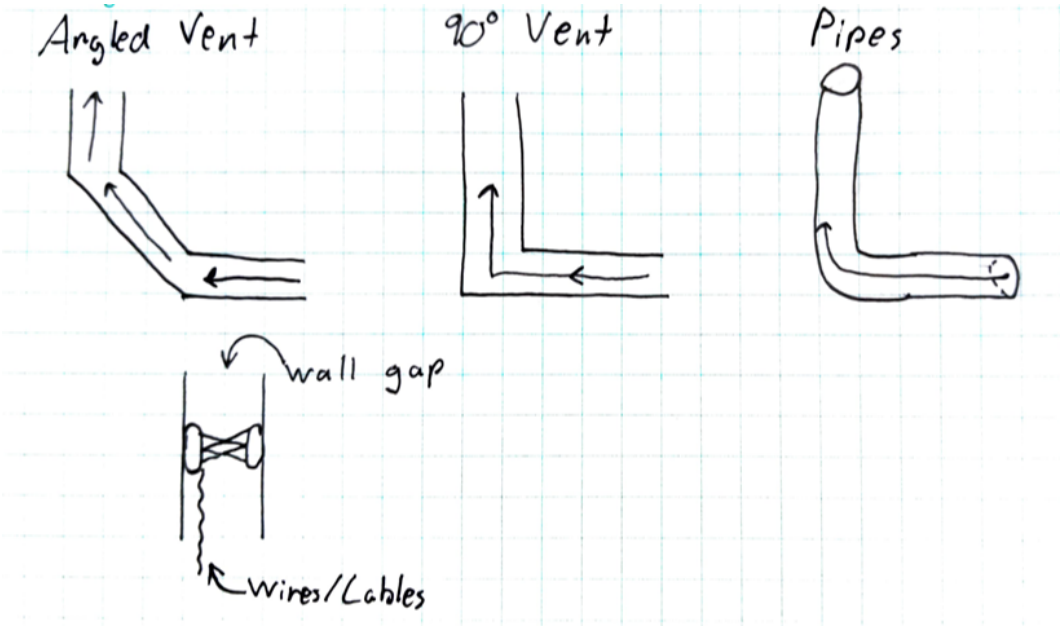


Figure 3: Sketches of Use Cases for Drone

-A number of scenarios that we plan for our drone to operate in. Bottom picture depicts the drone threading wires/cables through a wall.

1.4 High Level Requirements

- The drone should be able to complete an in-place 360 degree spin in addition to driving forward and backward as well as being able to climb at a 90 degree angle relative to the ground in tight vertical spaces.
- The top track should be able to expand at least three additional inches to fit the diameter of the space to apply additional traction.
- The drone should be able to drag and feed a wire up to 0.5 inches in diameter behind it while traversing a space.

The first requirement encapsulates all of our movement functionality. Spinning and moving forward and backward indicate that we've fulfilled all two-dimensional movement ability, while climbing at the 90 degree angle is the peak of our three-dimensional capability. The requirement is important for our overall purpose because coming up short in any regard means that there will be other, more appealing drone options.

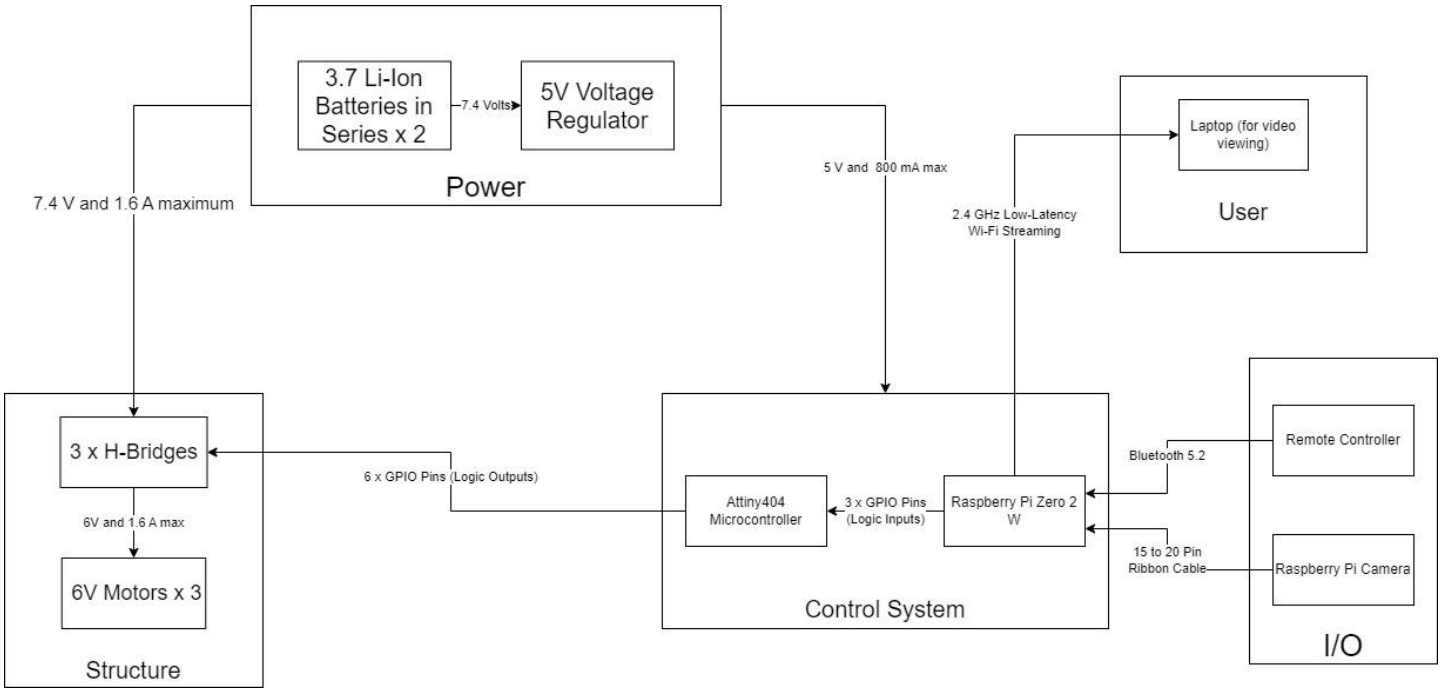
The second requirement needs to be met because having too short of an expansion arm means that our drone's primary advantage is not being leveraged

sufficiently. Three inches is a good portion of the overall length, and is nearly the maximum we can hit before the arm actually becomes a limiting factor in agility.

The third requirement is necessary because it's one of the goals, although a smaller one, that we set out to directly address. This functionality of being able to use the drone to feed wires in difficult spaces solves a widespread issue on its own. By having it as one of our high-level requirements, we are aiming to solve this challenge as part of the overall purpose of the drone.

1.5 Block Diagram

Figure 4: Block diagram of our subsystems.



Only minor changes were made to the block diagram over the course of the project. The biggest visual changes are largely just restructuring, as no components have been removed. The other notable change is that we changed the original 3.3V specification that led from the power subsystem to the control system to a 5 V requirement. While 3.3V was within operating capability for the Pi and the Attiny404, 5 V was necessary for programming as well as outputting 5 V on the General Purpose Input/Output (GPIO) pins. The H-bridges would require at least 5 V to recognize the logical input as a HIGH value, and the GPIO pins can only output equal voltage to the V_{in}

value on the microcontroller, so it was best to just change the voltage regulator to be the 5 V variant of the LM1117.

1.6 Subsystem Overview:

Power: The power subsystem's purpose is to supply the two halves of our project with the appropriate voltage and current necessary for operation. We used two, 3.7 Li-Ion batteries wired in series to provide a total of 7.4 V. The two branches we are powering are wired in parallel. One branch goes through a linear voltage regulator to bring the 7.4 V down to 5 V, which supplies our control system block along with providing a maximum of 800 mA. The other branch remains at 7.4 V and will draw no more than 1.6 A. This branch goes to the H-bridges that are connected to the motors.

Structure: The structure subsystem contains the mechanical components of our project. This means that the heart of this subsystem is our motors. There are 3, 6V motors that provide 5.5 kg-cm of torque, which we'll discuss later in the design and calculations section. These are controlled via the H-bridges that will output 6 V and a range of 100 mA when there is no load, to 1.6 A when the motors stall. This subsystem receives power from the power subsystem, and logic signals from the control system. In turn, it provides all the movement for our drone.

Control System: The control system is composed of two components, the Raspberry Pi Zero 2 W, and the Attiny 404 microcontroller. The inputs to the system as a whole are 5 V and 800 mA (at an absolute maximum) from the power subsystem, as well as a video data stream and Bluetooth remote control signals from the I/O subsystem. The Raspberry Pi receives both the video data stream and the Bluetooth signals. The video data stream is then sent over 2.4 GHz Wi-Fi to a viewing device and the Bluetooth signals are used as inputs for a program that will output three logic signals from the GPIO pins to the Attiny404. The microcontroller then uses those three logic signals to output 6 more logic signals to the H-bridges for motor control.

I/O: The I/O subsystem is composed of the Pi's camera and a Bluetooth remote controller. The camera is connected to the Pi via a ribbon cable, and transmits video over this cable. The Bluetooth controller transmits event codes for button pushes wirelessly to the Pi for it to process the meaning of those button pushes. These two components receive no inputs in return.

User: The user block demonstrated above is not technically a subsystem of the project, but is important to show in relation to the other, functional blocks of the project. The user needs to provide a device capable of connecting with the Pi wirelessly in order to view the live streamed video. Most likely this is a laptop that will SSH into the Pi's host network, then run a single command that starts the receiving of the video stream.

2 Design

2.1 Subsystem Design Process

In the following design sections, we've detailed the design process for each block, along with their associated requirement and verification tables.

2.1.1 Structure

The structural subsystem consists of the mechanical and physical aspects of the design. The main considerations of the structural system are the weight of the chassis, the power of the motors, and the physical dimensions of certain parts that allow us to fulfill our high-level requirements.

Design Decisions:

Firstly, the overall design of the drone came from many meetings with the machine shop. The initial concept was to use a single layer of material for the chassis, and to have scissor lifts that changed the height of a top-mounted track. There were a number of problems with the top-mounted track. First, the machine shop could not fabricate scissor lifts near the scale that we required in the time that they had. Second, the top track was supposed to be powered, and having two connection points as well as a motor was going to be very difficult, if not impossible given the aforementioned constraints. To this end, we negotiated a number of different options. The options were:

- An unpowered track on top operated by two rotating arms.
- Two linked arms that had wheels at the end, rather than a track.
- Two unlinked arms with wheels.
- A single arm, centered on the drone, also ending with a wheel.

The decision was made to choose the last option, as we realized that all we required was a method to get sufficient downforce to keep our drone grounded. A powered traversal method and a large surface area were not required. Therefore, we changed from the proposed top-mounted, powered track to an unpowered single arm-wheel piece with a three-inch radius from the tip of the wheel to the connection point on the chassis. Furthermore, after more consideration and suggestions from the machine shop, we opted for a two layer chassis, where we could sandwich the batteries between the layers, freeing up lots of space for the Pi and PCB to be mounted on the top. Lastly, we also had to determine the placement of the motors. If they were both on the front or both on the back, it would make us need a wider drone and cause some logistical problems for how we would wire the motors. Since this was a tracked drone, and not a wheeled one, we needed to ensure that we still had one motor on each side of the drone, so we opted for staggering the motors: one in front on the left, and the other on the right in the rear. This gave us the minimal width for our drone and made it easiest to wire them at the end of the project.

Another design process we went through was choosing our motors. Initially, we agreed that the smaller we could find, the better. Having already researched a similar type of drone [1], we found that Pololu micro-metal gearmotors would offer us the type of motors ideal for our project. From there, we decided on brushed motors over brushless, as the advantages of brushless (better lifespan and performance) did not justify the large difference in cost. The next decision was selecting which RPM motor we wanted to use. There was a large selection of motors available to us, so we had to figure out how much torque we thought was a safe amount for our requirements as well as if the required RPM to get said torque was fast enough to not be frustrating to operate. To this end, we selected the 84 RPM gearmotors with a 5.5 kg-cm torque ratio. The drone having 84 RPM was a good speed, as it was not too fast and not too slow either. It was quick enough to be very responsive, but not so quick that the user had to be careful about it being jumpy. Our final speed was ideal for a drone that needed to be careful in tight spaces.

The last design choice was about weight. We wanted the lightest drone possible, but it also needed to be reasonably sturdy. Therefore, we decided that aluminum was a good choice. Using aluminum meant that our drone's bare chassis

with the tracks and wheels weighed in at 0.66 lbs, which is quite light. However, we made sure to run the calculations on the motors and make sure that we'd still be within climbing ability. What follows is equation 2.1, which we used to calculate the weight that each motor will carry given the 5.5 kg-cm torque ratio and the radius of the wheels in the tracks:

$$W = T/r \tag{2.1}$$

$$W = \frac{5.5\ kg * cm}{\left(\frac{35}{2}\right) mm}$$

$$W = \frac{(5.5\ kg * 2.20462 \frac{lbs}{kg}) * cm}{1.75cm}$$

$$W = \frac{12.12541\ kg * cm}{1.75mm}$$

$$W = 6.9288\ lbs$$

In equation 2.1, *W* is the weight, *T* is the torque ratio, and *r* is the radius from the motor to the edge of the wheel. By solving equation 2.1, we can see that each motor is capable of pulling nearly seven lbs, well above the weight of the drone. However, we also needed to consider the additional weight of possible strings, cables, and wires. We believed that the excess weight capacity allowed potential users a lot of freedom in the distance and type of cable that they chose to lay down.

Finally, we elected to provide full power to the motors, meaning that they would have access to the full 6 V and up to 1.6 A if the load required it. We tested that they would run on other batteries as well, for instance we tested the motors ability to run on just one of our batteries should one of them die. The motors continued to provide sufficient pulling ability while running on 3.7 V. We also tested operation on just a standard AA battery. The motors ran noticeably slower, but were still operational. Attached in Appendix C, Figure 7 is a chart depicting the torque output curves based on amperage, voltage, and other factors.

Structure RV Tables:

Below, we've specified the requirements and verification methods for the structure subsystem:

Table 1: Structure Subsystem RV Table

Requirements:	**Verifications:**
Motors will operate off of 6 V and have current range from 100 mA (no-load) to 1.6 A (stall current).	• Attach motors to power system • Use a multimeter to measure both voltage and current when no load is attached. • Slowly add a load until the motor has stalled, being careful not to keep the motor stalled for too long. • We should see 6 V in both cases, 150 mA in the no load case, and then an increase in current draw while we add load until we maximize current at 1.6 A.
Expansion mechanism should add at least 3 inches.	• Manually extend the motor to its maximum height and measure the arm's radius.
Wire holder can hold wires up to ½ inch diameter.	• Measure the diameter of our holder • Feed various cables of diameter up to ½ inch and secure them • Ensure the wires remain secured in normal operation

Verification results:

The first requirement's verification yielded the results that the motors were drawing about 5.98 V to 6.03 V consistently. As far as the current draw went, we had about 180 mA under no load, which we realized wasn't quite true. The resistance of the axles and the treads that the motors were rotating was a very small load. They were smooth and offered very little resistance, but it was there, and it increased our "no-load" current by a very small amount. When the motors stalled, we measured about 1.55 A, which is less than we thought, but not a problem whatsoever.

The second requirement was easily verified. We just measured the arm that was mounted to the top of the drone. From the base of rotation to the tip of the wheel was 3.23 inches, surpassing our three inch requirement.

The third requirement was also easily verified. We measured the ring we attached to feed wires through at the rear of the drone, which had more than ½ inch clearance from the pylon it was attached to. This meant we could easily thread a cable of ½ inch diameter through the wire loop.

2.2.3 Power

The power subsystem consists of two, 3.7 V lithium ion batteries wired in series to provide a total of 7.4 V. Our motors present the highest voltage requirement at 6 V, so 7.4 V is more than enough to supply them. Parallel to the branch that supplies the motors is the line that sends 5 V to the control system, which is the other half of our design. In order to get 5 V, we use an LM1117 linear voltage regulator that sends a fixed 5 V.

Design Decisions:

The design of the power system was fairly straightforward. Lots of Li-Ion batteries came in 3.7 V, and were the appropriate size for the scale of our project. We decided at this point to simply get two and wire them in series to create a total of 7.4 V. The H-bridges we picked had built-in power regulation functionality, although it was limited. Despite the limitations, we didn't need a large amount of correction to go from 7.4 V to 6 V, so this was more than acceptable. Then, we needed to figure out how to get from 7.4 V to 5 V on the other side of the circuit. The LM1117 seemed like an easy, available solution, so we selected that. When it came time to order them, it was discovered that the LM1117 had a multitude of variants, and that one of them was a 5 V fixed output version. The decision was easy, and eliminated some of the forecasted difficulty with power system control. Lastly, when it came time to order the batteries, we knew we wanted something with a flat profile that could easily be fixed to the chassis. Looking through the list of 3.7 V batteries revealed that flat, square batteries existed as opposed to the classic cylindrical shape. At this point, we simply selected the batteries with the highest amp-hour rating that didn't break the profile of our drone.

Power RV Tables:

Table 2: Power Subsystem RV Table

Requirements:	**Verifications:**
Power subsystem should be supplying our H-bridges with enough voltage for them to output 6 V and 1.6 A maximum.	• Using a multimeter, check the incoming voltage for the Vin pin on each H-bridge • This verification step goes hand in hand with motor verification in structure
Subsystem should simultaneously be providing at least 5 V and up to 800 mA to the Pi, microcontroller, and logic elements of our circuit.	• Use a multimeter to ensure that we get 5 V, and that 800 mA or less is being properly drawn from the batteries.

Verification Results:

The first requirement of the power subsystem is the mirror image of the structure subsystem, so our results for both are the same.

The second requirement involved using a multimeter to verify that we were reaching the proper voltage thresholds. The actual measured voltage was consistently 5.024 V coming out of the linear voltage regulator, as well as across the terminals of the Attiny404 microcontroller and the pins used to feed the Pi for power. At no point did we ever manage a current draw greater than 314 mA, which meets our second requirement. This is about what we anticipated, but our circuit is prepared to output up to 800 mA.

2.2.4 I/O

The I/O system is fairly straightforward. It's composed of two components, the remote controller and the camera, both of which provide their outputs to the Raspberry Pi, which handles our wireless communication. The Pi functions as a bridge between the I/O and control subsystem, and more details on its operation will be located in the control subsystem section.

Design Decisions:

When choosing our camera, our options were fairly limited, making picking a specific model fairly straightforward. We chose one of the most basic cameras available, since we didn't need special features like IR video or 4K quality, both of which raised the price significantly. The bluetooth controller was just something we had on hand, and one of our team members already had experience with using a Dualshock 4 Playstation controller and pairing it with a Raspberry Pi. The controller would provide an intuitive way to control the drone for a user of any experience level with remote controls. Both components of our I/O subsystem simply had to be connected to the Pi, either physically or digitally. There was initially an issue with finding the correct camera commands, as the documentation we were using was outdated [5]. However, we discovered soon after what the commands had changed to. Only one keyword had changed.

I/O RV Tables:

Table 3: I/O Subsystem RV Table

Requirements:	**Verifications:**
The Pi Zero should transmit video to a viewing device over Wi-Fi.	• The user should attempt to connect with the Pi Zero over Wi-Fi • A live video stream should be viewable on the chosen viewing device.
The Pi Zero should receive commands via Bluetooth from a controller	• The drone should respond to commands given by the remote control in a way that accurately and quickly reflects the input given over a Bluetooth connection.
Mounted camera should transmit color video back to Pi via the wire harness meant for connection between the two, at which point the Pi passes the video onto the user's viewing device.	• Turn the drone on and connect the viewing device to the Pi over Wi-Fi • Once connection is established, we should have access to the live feed in color and without unmanageable latency

Verification Results:

All three requirements were verified the same way, in that upon running the associated programs for all three functions, we should be able to visually verify the functionality of these requirements. For the first and third requirements, we were able to view low-latency, color video via a Wi-Fi connection on a team member's laptop, meeting the verification for both of those. The second requirement was also met, as we both witnessed and recorded proof that our drone was responding to real-time inputs to the bluetooth controller.

2.2.5 Control System

The control system consists of two components, the Pi Zero and the Attiny404. The Zero uses the camera and the bluetooth controller as input. From there, it is used as an access point where an external device may wirelessly connect with it so that it can stream a live video feed to this external device. The input it receives from the bluetooth controller is parsed using existing libraries on the Pi into usable data. That data is then used in turn in a short program that creates outputs on three of the Pi's GPIO pins that are sent to the Attiny404. The Attiny404 takes these three inputs, and outputs six logical signals based on a truth table we crafted. Those six logic signals are sent to our H-bridges for motor control. The truth tables can be followed from Pi to microcontroller to H-bridge using Table 7, 8, and 9.

Design Decisions:

At first, we were going to use the Raspberry Pi Pico, which is a smaller version of what we have. However, as we went through the initial design stages, we found that the Pico was lacking in other areas, such as video. It would have taken quite a bit more work than just plugging in our camera module to get video running on a Pico, so we opted for the Zero 2 W.

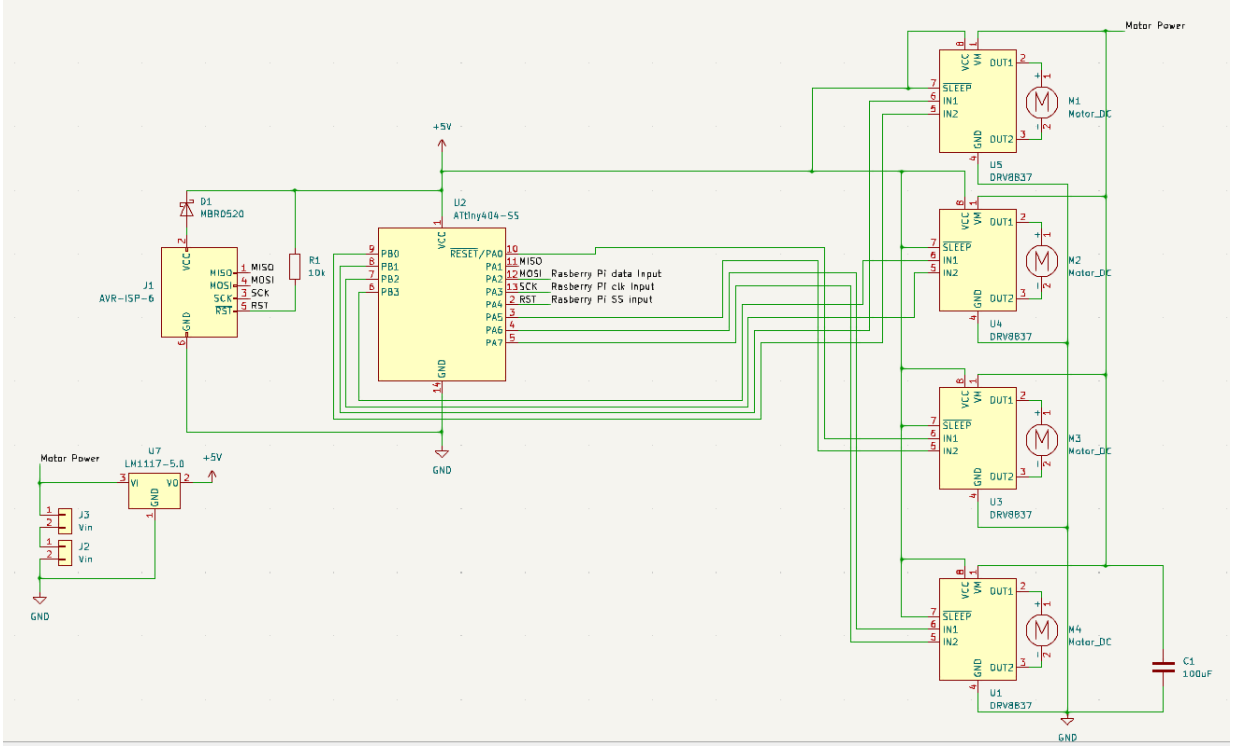
Our first iteration for the PCB design involved using an Attiny85. We didn't require lots of power, just something that was going to be suitable for running a simple input/output program. In conjunction with the Attiny85, we planned on having an 8-bit serial register that would act as a GPIO expander. It would take in serial data with SPI, then output 8 bits in parallel when it received the appropriate clock signal. We wanted to stay with something we were at least close to familiar with, so we found the Attiny404, which had more than enough GPIO pins for our

uses. At this point, we simply wired the H-bridge inputs directly to the microcontroller.

We would have further issues with the microcontroller however. Late in the design process we discovered that the Attiny404 was actually set to program with UPDI by default, not SPI. This meant we needed to have a breadboard and set up an Arduino as a temporary programmer using a jtag2updi program we found online [6]. Furthermore, we found out that not all GPIO pins could output the full 5 V necessary for logical high for our H-bridges. Specifically, only the pins rated as Analog GPIO were suitable. Of the three pins that were connected to non-Analog GPIO, Two of the three were also SCLK pins, and led to a single H-bridge. Since we had four bridges and only three motors, we could write this off as a loss and continue unimpacted. The last of the three pins was actually the UPDI programming pin, and only 3.3 V capable for output. We ended up bending this pin upward so it didn't touch its trace pad on the PCB, and rerouted the last unused 5 V capable GPIO pin to replace it using a wire we soldered on. We only had nine GPIO pins that were 5 V capable. We needed six of them for outputs, and three for inputs, meaning that we didn't have any spares, hence the decision for the wire and bent leg.

Lastly, we ended up using the six pins that our SPI programmer would have used as entirely different uses altogether. Since SPI wasn't going to work to program our board, this was alright, or even advantageous since we had more flexibility to work with. The programmer had a 5 V pin and a GND pin, which we soldered wires into to create a power loop for our Pi to be plugged in, as opposed to creating parallel wires to leech these two values from. The three of these pins were already being used as the three GPIO input traces from the Pi, so we soldered wires into the through holes as opposed to the solder pads we initially planned on using. The last remaining hole was where we soldered the wire from the initially unused GPIO pin, since it led to the pad that the UPDI programming pin was now bent upwards from. This made the improvised solution much easier to implement, as we didn't need to solder from the GPIO pin to the small microcontroller pad, but instead a through hole that was on the same circuit. See Figure 5 for the schematic of our control system.

Figure 5: Control System Schematic



Control System RV tables:

Table 4: Control Subsystem RV Table

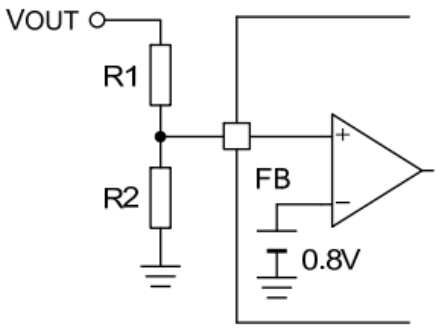
Requirements:	**Verifications:**
The Attiny85 will receive a code from the Pi that specifies a motion type to carry out and will send the appropriate bits to the H-bridges	• Based on our truth table in Appendix C, Tables 7, 8, and 9, we should be able to measure the IN1 and IN2 pins on each H-bridge to make sure they align with what we expect to see based on the input we give.

Verification Results:

During testing, we verified the proper operation of the control subsystem by using our multimeter to measure the voltage on the proper GPIO pins that lead to the IN1 and IN2 pins. Using the code we had written as a guide, we checked that for every input from the Pi, we received a logical high or a logical low according to the truth table/code. At first we did receive some issues, but the end result had all outputs faithfully matching the truth table we had written into the code.

2.2 Tolerance Analysis

Since we're using the voltage regulator mentioned above to control the input to the motors, it's a critical component to the whole design. The datasheet [3] contains some warnings and suggestions that talk about our choice of resistances for our output. Specifically, we need to hit a target feedback ratio if we want to ensure the proper V_{out} value of 6 V.



$$V_{OUT} = \frac{R_1 + R_2}{R_2} \times 0.8 \text{ [V]}$$

※ Minimum pulse range that can be produced at the output stably through all the load area is 200nsec for BD9E302EFJ.
Use input/output condition which satisfies the following method.

$$200(nsec) \leq \frac{V_{OUT}}{V_{IN} \times F_{OSC}}$$

Figure 74. Feedback Resistor Circuit

Please set feedback resistor R1 + R2 below 700 kΩ . In addition, since power efficiency is reduced with a small R1 + R2, please set the current flowing through the feedback resistor to be small as possible than the output current I_O.

Figure 6: Circuit Diagram for use of linear regulator

Here, we know we need to have a combination of R1 + R2 that does not exceed 700 kOhms, but we also need to find a valid combination that is large enough to make sure our power efficiency is not affected. The math would be as follows:

V_{out} = 6 V

6 = (R1 + R2)/ R2 * 0.8
7.5 = (R1 + R2)/R2
7.5(R2) = R1 + R2

From here, we actually get to decide what we think is a valid solution for R1 and R2, so we can put in a value for R2 and see what R1 would need to be. After some guess and check, we found that we can use the 10 kOhm ERJ-PB3B1002V with a 65kOhm

ERA-3AEB6492V resistor. Both of these can be found in a $\pm 0.1\%$ variant, so if we run all four possible cases:

<u>+0.1% (10kOhm) R2 and +0.1% (65kOhm) R1</u>

$$(10,010 + 65,065)/(10,010) * 0.8 = 6\,V$$

<u>+0.1% (10kOhm) R2 and -0.1% (65kOhm) R1</u>

$$(10,010 + 64,935)/(10,010) * 0.8 = 5.9896\,V$$

<u>-0.1% (10kOhm) R2 and +0.1% (65kOhm) R1</u>

$$(9,990 + 65,065)/(9,990) * 0.8 = 6.0104\,V$$

<u>-0.1% (10kOhm) R2 and -0.1% (65kOhm) R1</u>

$$(9,990 + 64,935)/(9,990) * 0.8 = 6\,V$$

As we can see above, we have very small deviations in voltage that fall within the range of acceptable voltage values for our motors, so there won't be any disruption of performance from the motors. This falls well within tolerances with easily obtainable parts.

3 Cost Analysis and Schedule

Cost Analysis

Table 5: Cost Analysis Table

Item	**Quantity**	**Price**	**Total**
2 MP Camera	1	$24.99	$24.99
Raspberry Pi Zero W	1	$14.99	$14.99
Tracks	1	$14.95	$14.95
Polulu Motors	4	$22.99	$91.96
H-Bridge Driver	4	$0.89	$0.89
Cable Clip	1	$0.89	$0.89
LithIO Batteries	1	$18	$18
PCB stencil	1	$46.39	$46.39
Item Subtotal			$213.06
Machine Shop		$200	$200
Labor	(10 Hours/week * 3 * 11 weeks)	$48.50/ hour	$16,005
Total			**$16,368.67**

Schedule

The schedule our team followed can be found in Appendix B, along with the work that each team member accomplished.

4. Conclusion

4.1 Conclusion

Our group was, at one point, able to successfully stream video and achieve full operation of our drone via remote control. Unfortunately, before we were able to rigorously test its vertical climbing ability and its demonstration, one of the pads we had soldered a motor control wire to peeled off of our PCB. That is to say, at one point we had full functionality. After we couldn't properly fix the broken component, we still had functionality of all motors besides the top expansion arm. This meant that video, remote control, and standard two dimensional traversal still worked.

Should we have to do this project again or provide suggestions for someone to pursue the same goal, we identified many sticking points, issues, and changes we'd make for the future. The first thing we would change or fix is the solder pads we planned on soldering our wires to. The initial plan was to connect the motors and the Pi using wires that we'd attach to our PCB with small pads. This was a mistake, and we'd remedy it by instead adding through holes to solder the wires through or solder in some pins for female connectors. The second issue is the scale of our project and PCB components. Many of the issues we had stemmed from the very small scale of our parts. The pads for the motor control wires were too small, and the gap between them left no room for soldering mistakes. The H-bridges were 2 mm by 2 mm, which also created issues. We'd account for human manufacturing limitations in the future. Lastly, we didn't consider cable management in the design for our drone. This isn't a hard problem to fix, but it should be considered for both practical and aesthetic reasons. Either a change in the chassis or by choosing more customized wires and holders would easily remedy this problem.

4.2 Ethics and Safety

In terms of ethics, our group followed IEEE Code of Ethics adopted by the IEEE Board of Directors through June 2020 [4]. We recognize that technologies have the ability to affect one's life thus we hold ourselves to the highest ethical standard when working professionally in a team. For full details on how we adhered to the IEEE Code of Ethics and our regard to safety in our project, see Appendix A.

Appendix A

Ethics and Safety

In terms of ethics, our group followed IEEE Code of Ethics adopted by the IEEE Board of Directors through June 2020 [4]. We recognize that technologies have the ability to affect one's life thus we hold ourselves to the highest ethical standard when working professionally in a team including but not limited to:

1. Seeking and providing truthful reviews and feedback of our technical work [4]

Within our group, we will follow course guidelines for timely feedback and confirmation with Teaching Assistant and Professors. We have already met with our Teaching Assistant once and plan on doing so, every week, for the remainder of the semester

2. Constantly learning and acquiring new skills throughout the training and design process [4]

We will also ensure to consult expertise (Professors, Teaching Assistants, Machine Shop Technicians) if questions or uncertainties arise during this project. This group consists of members with different areas of expertise, including power, programming, raspberry pi programming, PCB design, motors etc. By working together, we will be able to constantly learn from one another and ensure we all are successful.

3. Treating all people with respect and kindness and ensuring these codes are properly followed [4]

To ensure good teamwork and efficient communication, our team established a Discord server. With a shared Google Drive storage space, we made sure that all documentation is accessible for all team members. GitHub is used for software and schematic version control along with storage. This system not only keeps track of all technical details but also confirms that all members are on the same page.

As far as ethical concerns go for our project specifically, we've addressed them as follows:

1. All of our electrical components will be "stowed" so that they are not exposed and dangerous to the operator nor will they cause unintended damage to the small spaces that we plan on having our drone operate in.

2. We'll ensure to use the existing standard for Bluetooth and Wi-Fi connections for security. The connection will only be between operator and drone, with the transmitted data being used for no other purpose than meaningful guidance of said drone.

3. The motors and treads used in the design will be designed as safe as possible so our project won't cause unintended harm to an operator's fingers or other small objects. The motors should not spin excessively quickly or be pushed well beyond their safe limits.

Appendix B

Schedule

Table 6: Schedule

Week	Task	People
10/2 - 10/8	• Submit design plans to machine shop	• Jacob
10/9 - 10/15	• Heavily revised drone design based on feedback from machine shop • Order first round of parts • Review of PCB design	• Jacob • Jacob • Jeffrey
10/16- 10/22	• First PCB design submitted • Second round of parts ordered	• Jeffrey • Jacob
10/23-	• Met with machine shop again, design	• Jacob

10/29	finalized	
10/30-11/5	• Soldered all received parts on the board	• Jacob
11/6-11/12	• Requisitioned last remaining missing components from shop • Soldered last two pieces	• Jeffrey
11/13-11/19	• Finished and received chassis • Initial microcontroller code written	• Jacob • Jeffrey
11/20-11/26	• Remote control and camera streaming became functional • Pi GPIO tested, motors tested • Bought lots of spare parts • Attempted to program microcontroller	• Jacob • Jacob • Jacob • Jeffrey
11/27-12/3	• Microcontroller successfully programmed • First PCB thoroughly tested, partially working • Second PCB created, tested • Pi, PCB, Chassis integration • Project has full functionality	• Jacob +Jeffrey

Appendix C

Control System Truth Tables

Table 7: Movement Inputs vs Pi Output Codes

Movement	Pi Output Code
Brake	000
Forward	001
Backward	010
Lift Arm	011
Lower Arm	100

Left	101
Right	110

Table 8: Pi inputs vs Microcontroller Outputs

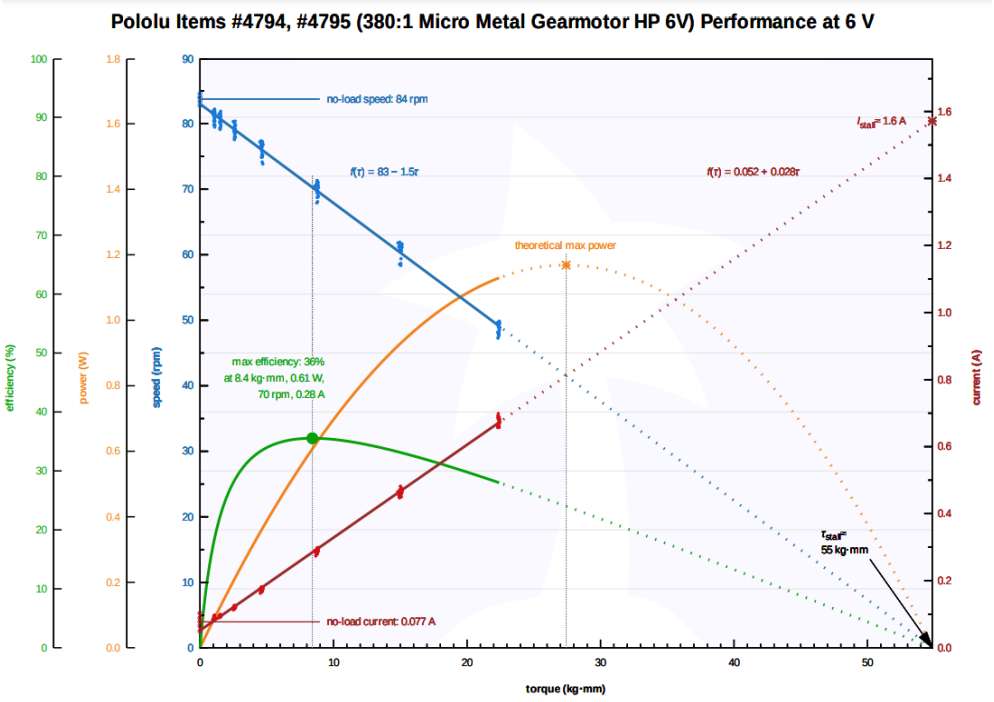
Microcontroller Input	Output (pairs of logic values for left, right, and top motors)
000	11 11 11
001	10 01 11
010	01 10 11
011	11 11 10
100	11 11 01
101	10 10 11
110	01 01 11

Table 9: H-Bridge Movement Methods and Input Codes:

	Left Motor (IN1, IN2)	Right Motor (IN1, IN2)	Top Motor (IN1, IN2)
Coasting	00	00	00
Forward	10	10	01
Rotate Left	10	01	xx
Rotate Right	01	10	xx
Backward	01	10	01
Brake	11	11	11

Motor Torque vs Efficiency, Speed, and Current Graph

Figure 7 - Torque vs Current, RPM, Power, and Efficiency Graph [2]



References

[1] O. Inbar and D. Zarrouk, “Analysis of climbing in circular and rectangular pipes with a reconfigurable sprawling robot,” *Mechanism and Machine Theory*, vol. 173, p. 104832, Jul. 2022, doi: https://doi.org/10.1016/j.mechmachtheory.2022.104832.

[2] Pololu Robotics and Electronics, “Pololu Micro Metal Gearmotors,” Pololu Corporation. [Online]. https://www.pololu.com/file/0J1487/pololu-micro-metal-gearmotors_rev-5-1.pdf (visited on 10/3/2023)

[3] *“7.0V～28V Input, 3A Integrated MOSFET Single Synchronous Buck DC/DC Converter* ”, BD9E302EFJ, Regulator, Sept 28, 2023

[4] IEEE. “”IEEE Code of Ethics”.” (2020), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 9/14/2023).

[5] “Raspberry pi documentation,” Camera, https://www.raspberrypi.com/documentation/accessories/camera.html (accessed Dec. 6, 2023).

[6] Warner, Teddy. “UPDI Serial Programming - TeddyWarner.org.” *Teddywarner.org*, teddywarner.org/Projects/SerialUPDI/#. Accessed 12 Nov. 2023.