UNIVERSITY OF
ILLINOIS
URBANA-CHAMPAIGN

Vertical Climbing Drone

Electrical & Computer Engineering

Team 40

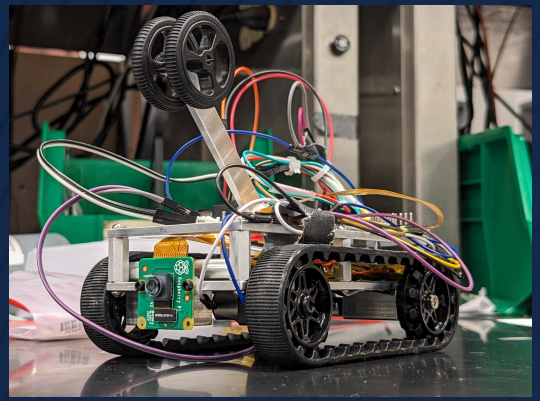
Jacob Corsaw (jcorsaw2) and Jeffrey Chang (jdchang3)

12/5/2023

Project Objective

Lots of remote control drones already exist that are similar to ours, so what sets our project apart?

VS.



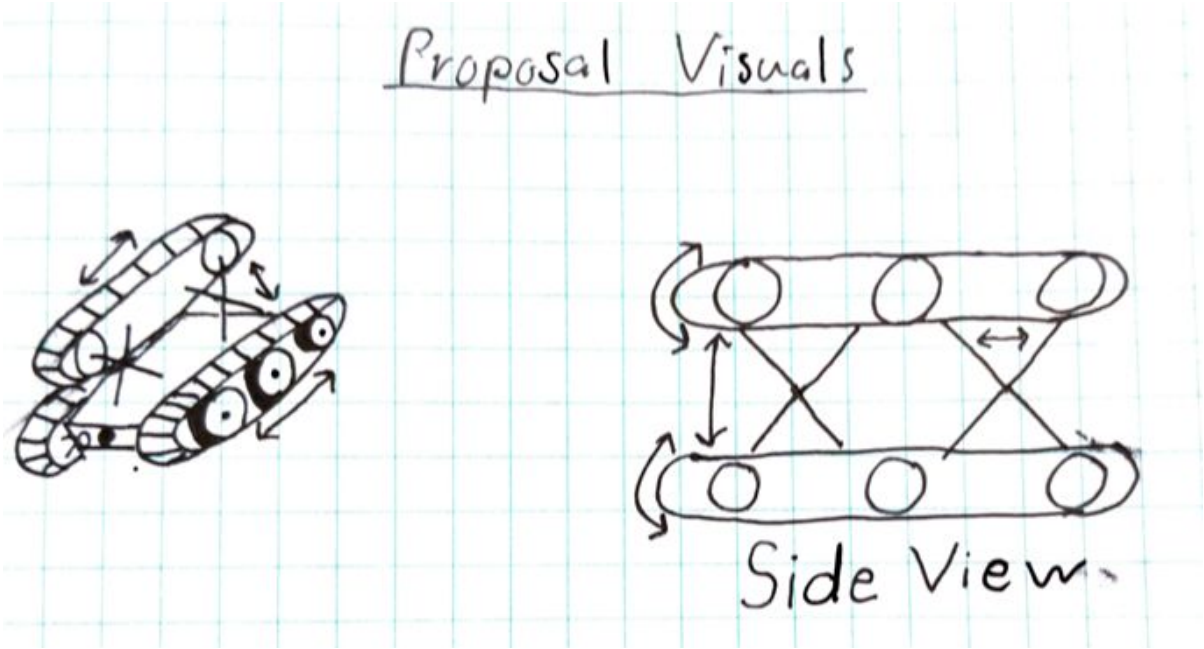
High Level Requirements

- 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 3 additional inches to fit the diameter of the space to apply additional traction.

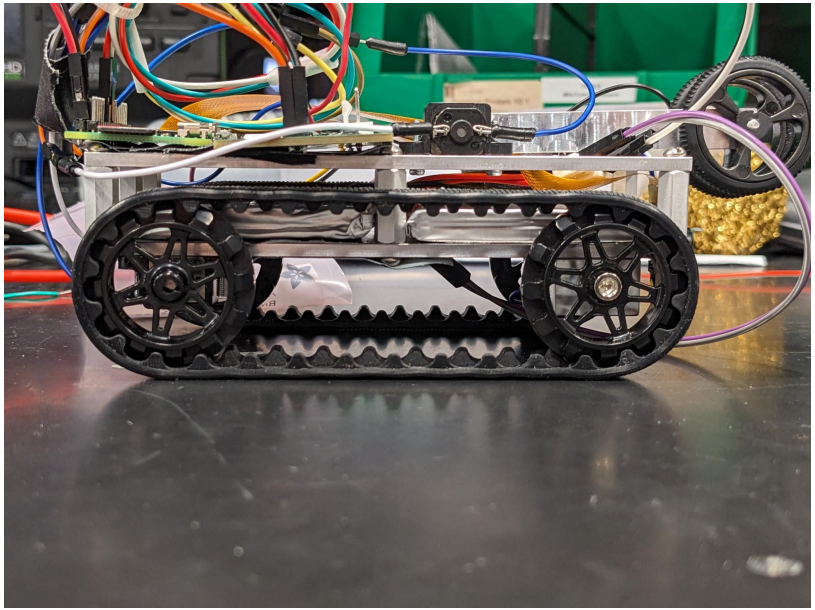
- The drone should be able to drag and feed a wire up to 0.5 inches in diameter behind it while traversing a space.

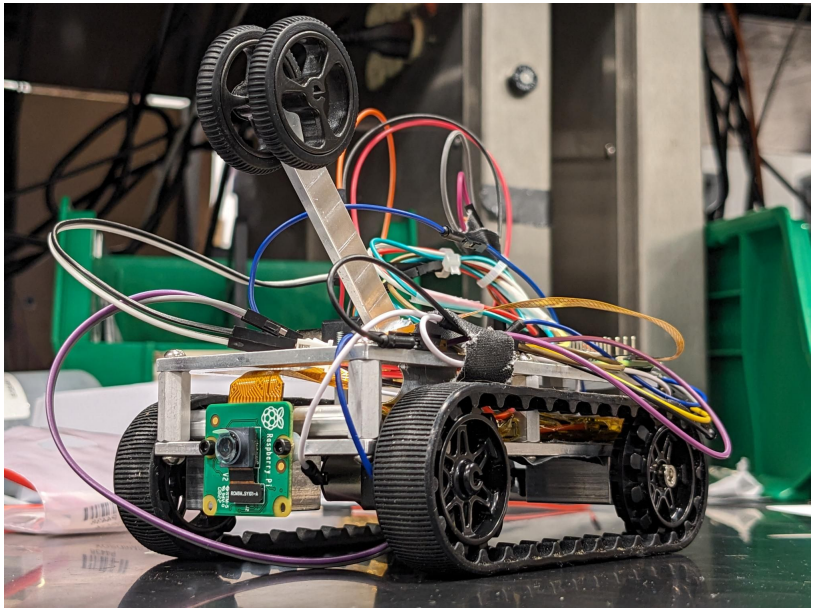
Original Concept:



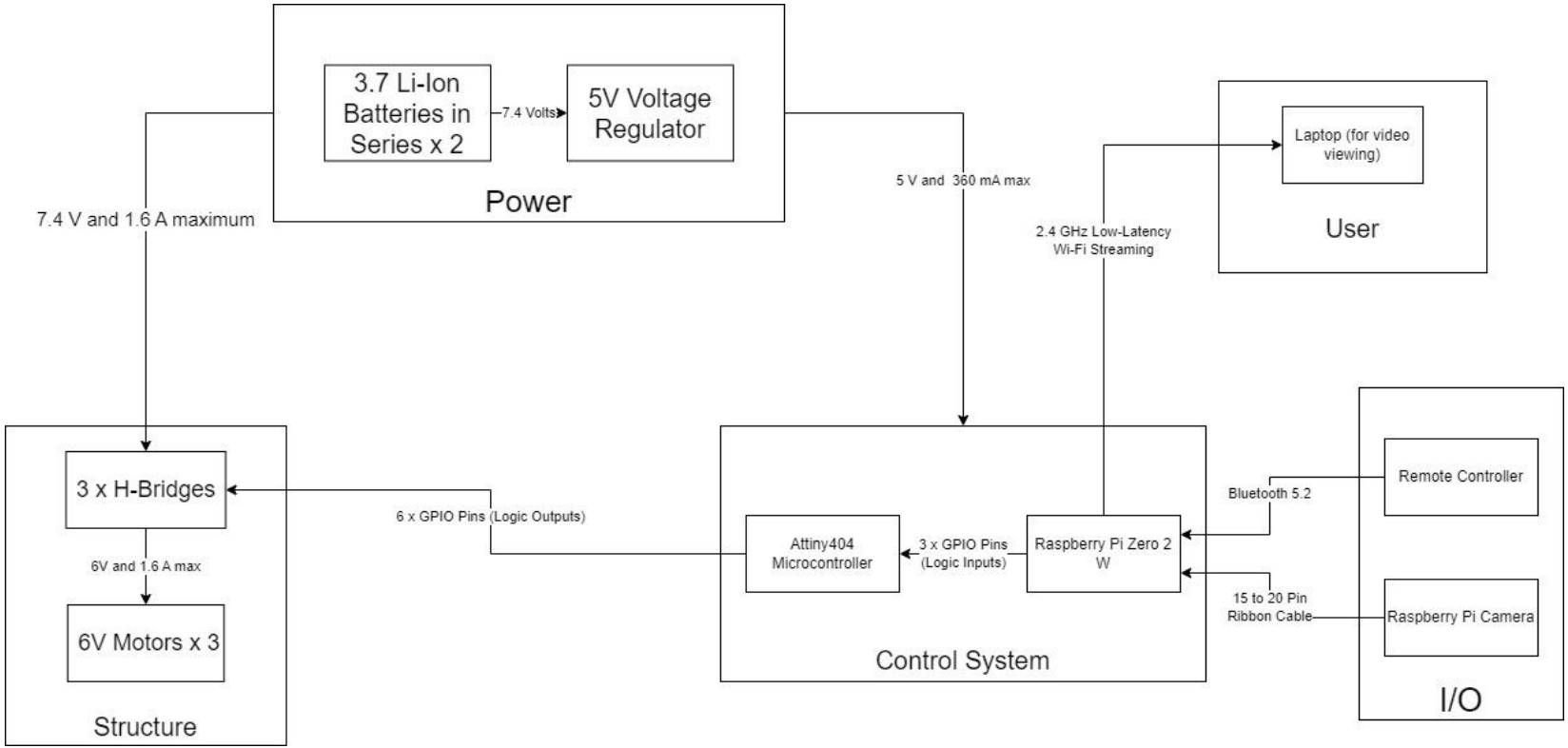
Original Concept vs Final Product

Final Product:





Block Diagram



Pi and Microcontroller Truth Tables

Pi Output Truth Table

Movement	Pi Output Code
Brake	000
Forward	001
Backward	010
Lift Arm	011
Lower Arm	100
Left	101
Right	110

Microcontroller Truth Table

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

Microcontroller and H-Bridge Truth Tables

Microcontroller Truth Table

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 7-1. DRV8837 Device Logic

nSLEEP	IN1	IN2	OUT1	OUT2	FUNCTION (DC MOTOR)
0	X	X	Z	Z	Coast
1	0	0	Z	Z	Coast
1	0	1	L	H	Reverse
1	1	0	H	L	Forward
1	1	1	L	L	Brake

Structure Subsystem

Requirement	Verification
Motors will operate off of 6V and have current range from 100 mA (no-load) to 1.6 A (stall current).	• Use a multimeter to check that the motors always have 6 V across the terminals, and that current begins at 100mA (no load) and increases to 1.6 A when they stall.
Expansion mechanism should add at least 3 inches.	• Measure the radius of our expansion arm to ensure it's 3 inches or greater.
Wire holder can hold wires up to ½ inch diameter.	• Measure the diameter of the space for the wires to ensure it's ½ inch.

I/O Subsystem

Requirements:	**Verifications:**
Mounted camera should allow the Pi to view low-latency full-color live video to the viewing device.	● Start the streaming program on the Pi and the viewing device. ● If we get near-live and in-color streaming, then it's a success.
The Pi Zero should transmit video to a viewing device over Wi-Fi.	● Using the Pi's Wi-Fi network, we should be able to SSH into the Pi on our viewing device and perform the test the same as above.
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.

Power Subsystem

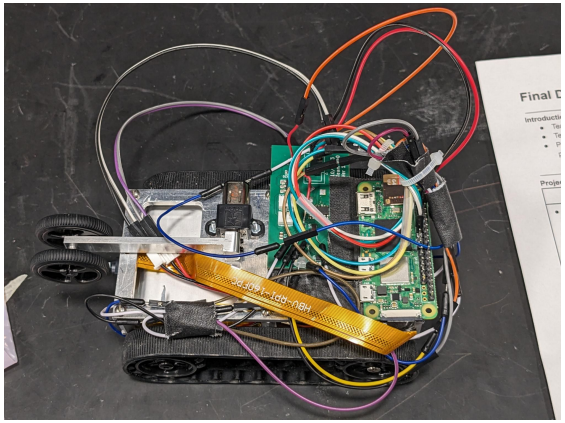
Requirements:	**Verifications:**
Power subsystem should be supplying our H-bridges with enough voltage for them to output 6V and 1.6 A maximum.	• Using a multimeter, check the incoming voltage for the Vin pin on each H-bridge
Subsystem should simultaneously be providing at least 5 V and up to 800mA.	• Use a multimeter to ensure that we get 5 V to our Pi and microcontroller, and verify that microcontroller and Pi operation is consistent, indicating sufficient current flow.

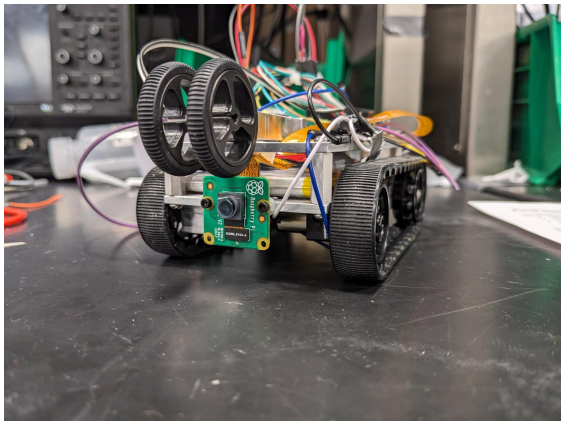
Control System

Requirements:	**Verifications:**
The Attiny404 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 below, 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.

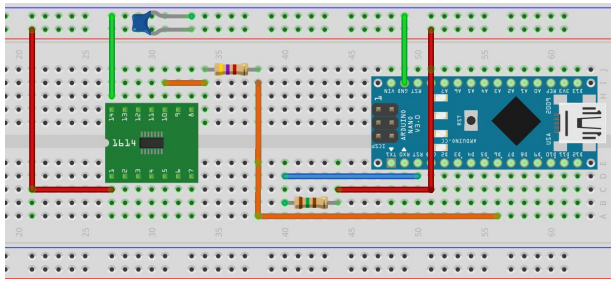
Results

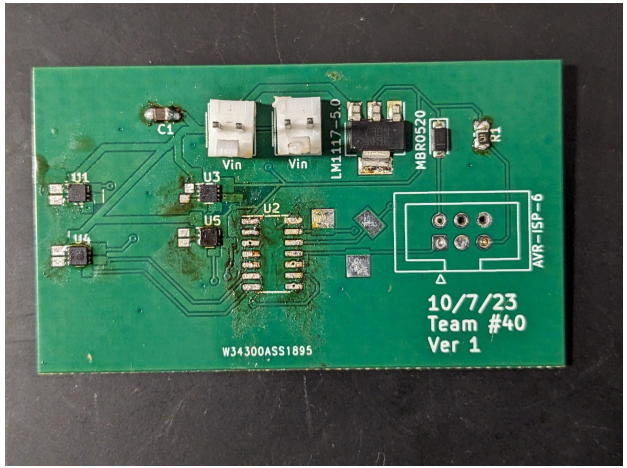
- All subsystem requirements were met.
- Verifications were either performed as we built each module on the PCB, or were inherently proven through proper operation of the component
- High level requirements were unfortunately not met, due to a last minute breakage.





Results (Video)





Issues and Solutions:

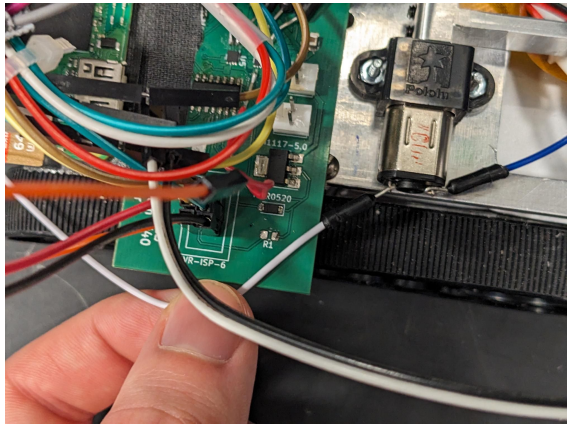
Microcontroller Programming:

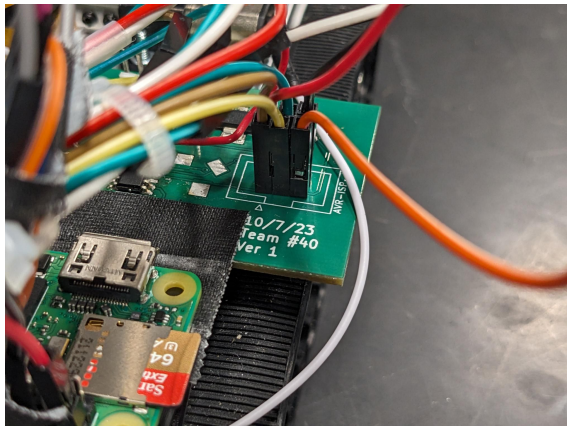
- Microcontroller used UPDI, not SPI
- Would not program under battery power

PCB Implementation:

- Solder issues
- Size
- Repeated solder/desoldering
- Excess heat damage

Development Process





Issues and Solutions:

2nd PCB Issues

- GPIO pins do not all provide 5V output
- Unorthodox workarounds
  - Removed resistor
  - Repurposed programmer pins
  - Created new contact
  - Bent microcontroller leg

Raspberry Pi Connectivity Issues

- Crowded signals caused connection resets

Solder Pad Fell Off

- One of our contact fell off entirely

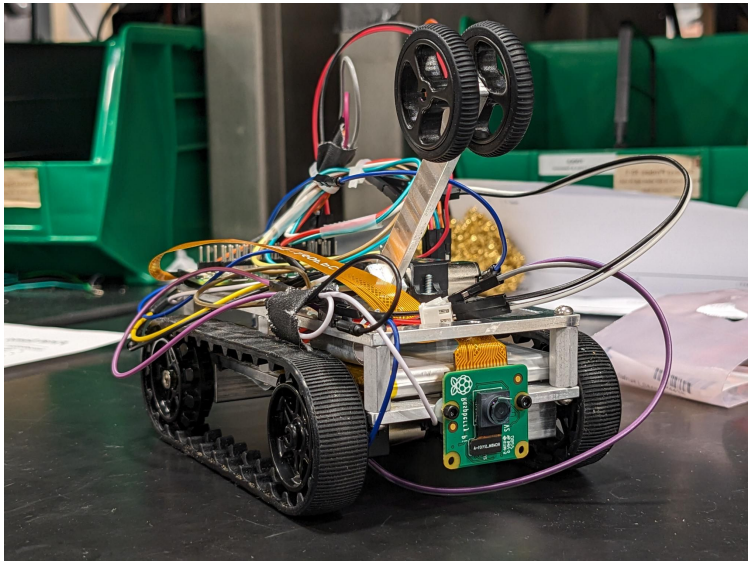
Conclusions and Lessons Learned

Lessons Learned:

- Illustrated the need to pay attention to scale in prototyping
- Logistics
- Designing in theory vs practice
-

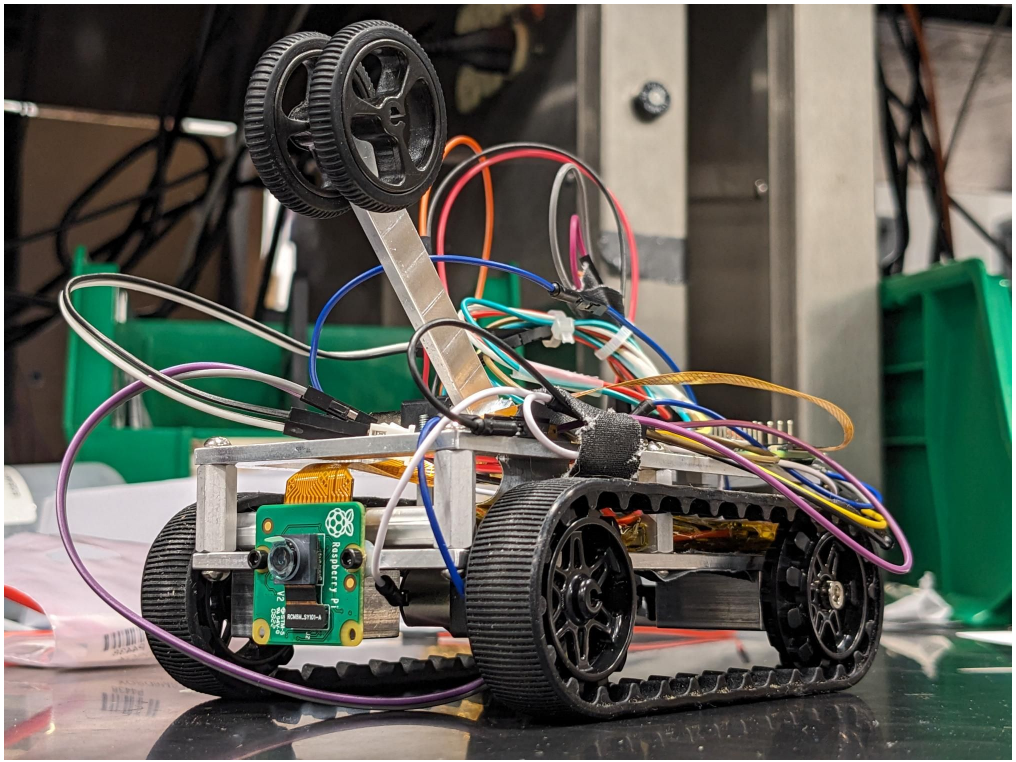
Changes with Hindsight:

- Much better at soldering
- Would have tested earlier and more often
- Different PCB design philosophy to make it easier to test and work with.
- Would have put more effort into integration aesthetics



Future Work:

- More robust network connection ability
- Redesigned PCB
- Updated chassis
- Custom connections



Thank You for Listening to our Presentation!

Questions?

Team 40:
Jacob Corsaw, jcorsaw2@illinois.edu
Jeffrey Chang, jdchang3@illinois.edu

The Grainger College of Engineering

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN