













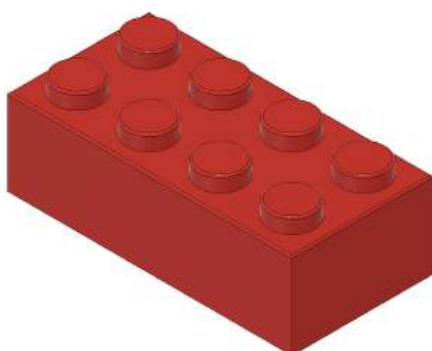
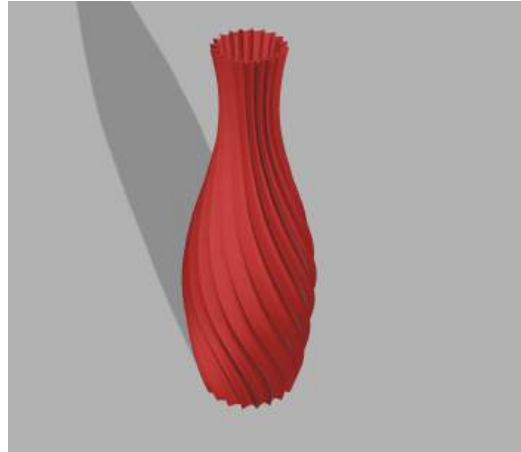
# Photos of the Humanoid Project

(and the process behind it)

# Chapter 1: Learning CAD

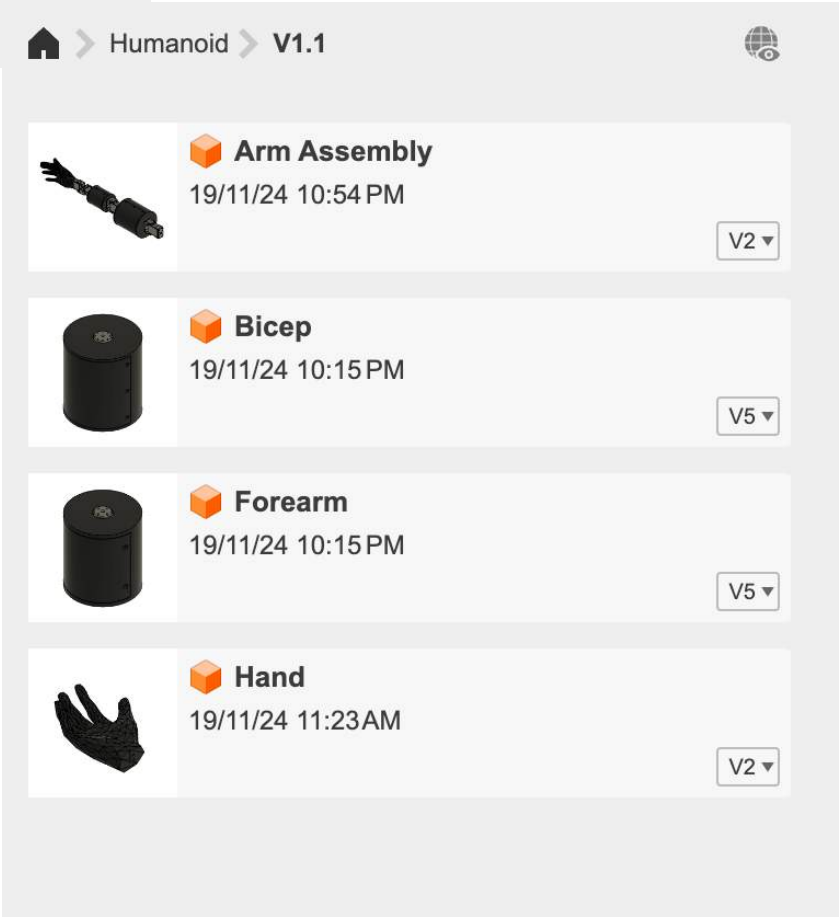
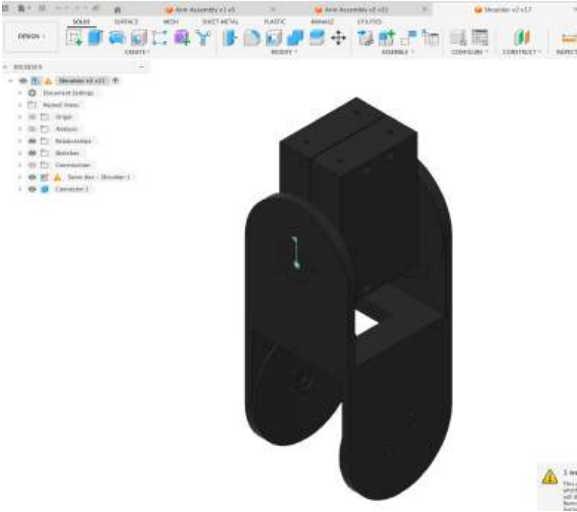
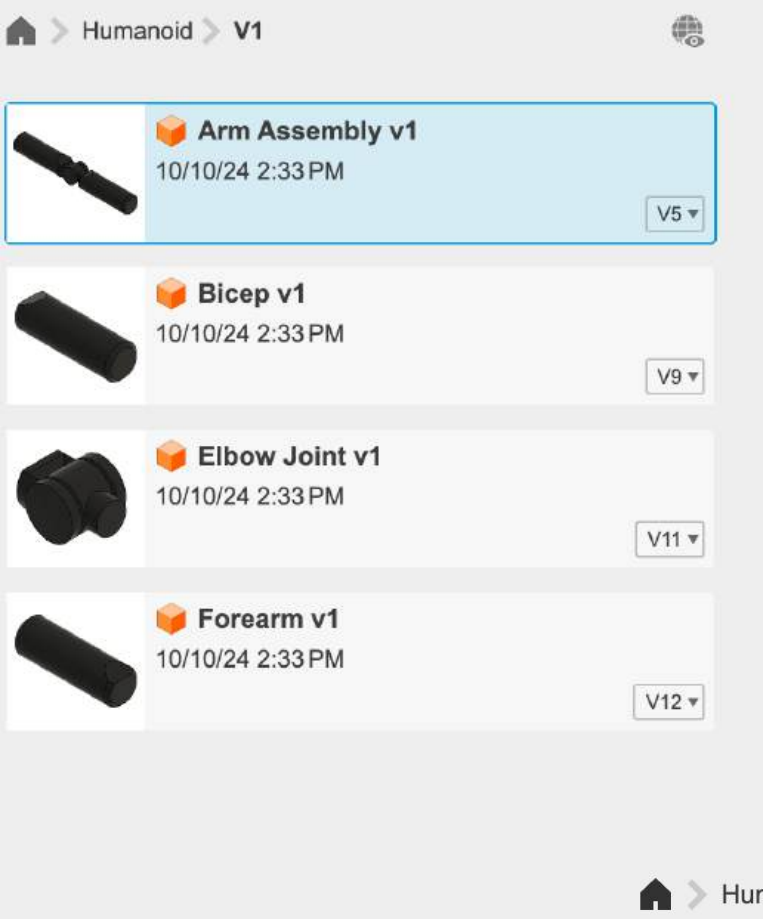
(September 2024)

	<b>15. Painter's Tripod</b> 18/09/24 2:06 PM	V2 ▾
	<b>16. Design Intent (theory)</b> 18/09/24 2:26 PM	V3 ▾
	<b>17. Sketch Constraints (theory)</b> 18/09/24 2:27 PM	V1 ▾
	<b>18. STL to Solid</b> 18/09/24 5:53 PM	V3 ▾
	<b>19. Box</b> 18/09/24 5:52 PM	V2 ▾
	<b>20. Hinge Box</b> 18/09/24 8:56 PM	V4 ▾
	<b>21. Keychain Mold</b> 18/09/24 8:30 PM	V3 ▾
	<b>22. Chess Mold</b> 18/09/24 8:55 PM	V2 ▾
	<b>23. Tripod</b> 21/09/24 12:47 PM	V4 ▾
	<b>24. Napkin Holder</b> 20/09/24 9:44 PM	V2 ▾
	<b>25. Propellor</b> 20/09/24 10:14 PM	V2 ▾
	<b>26. Raspberry Pi Case</b> 20/09/24 10:51 PM	










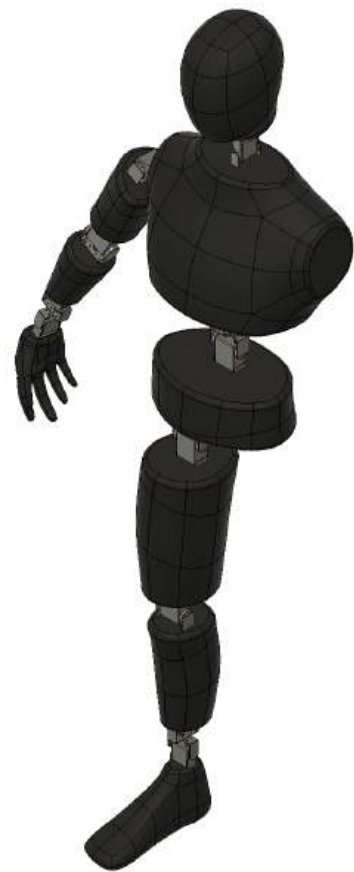
# Chapter 2: Early Designs

(October 2024)



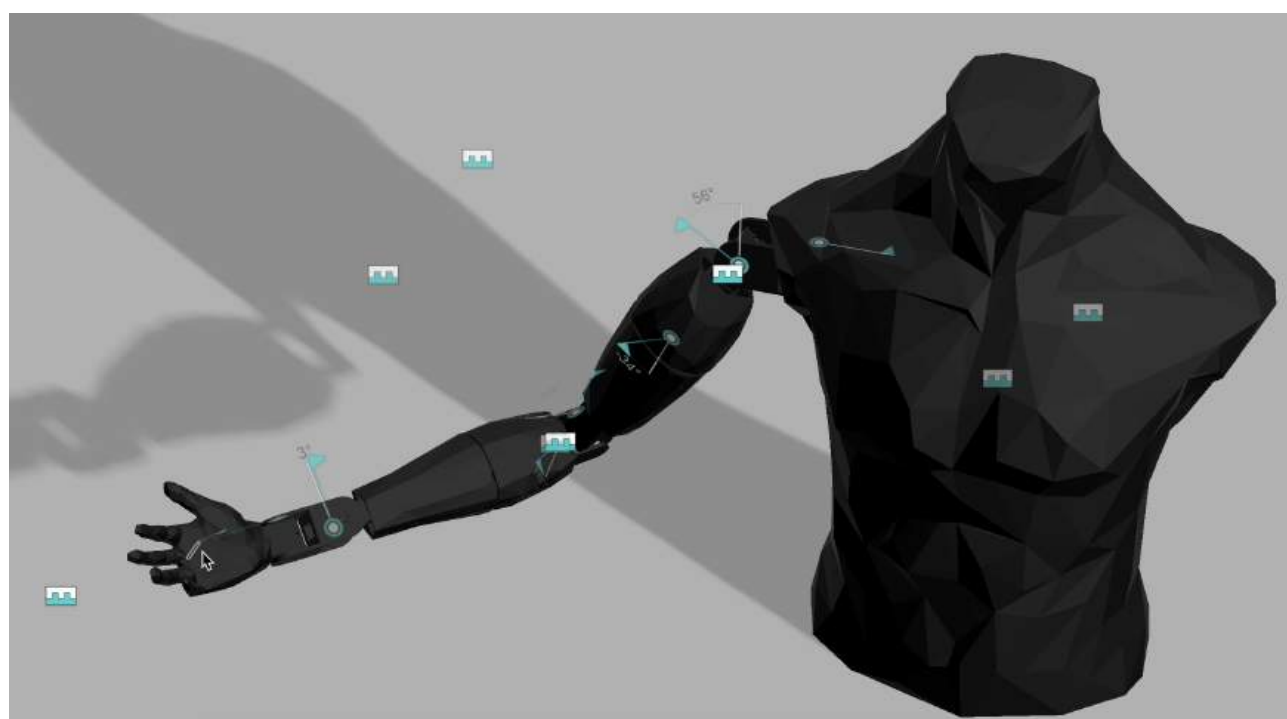
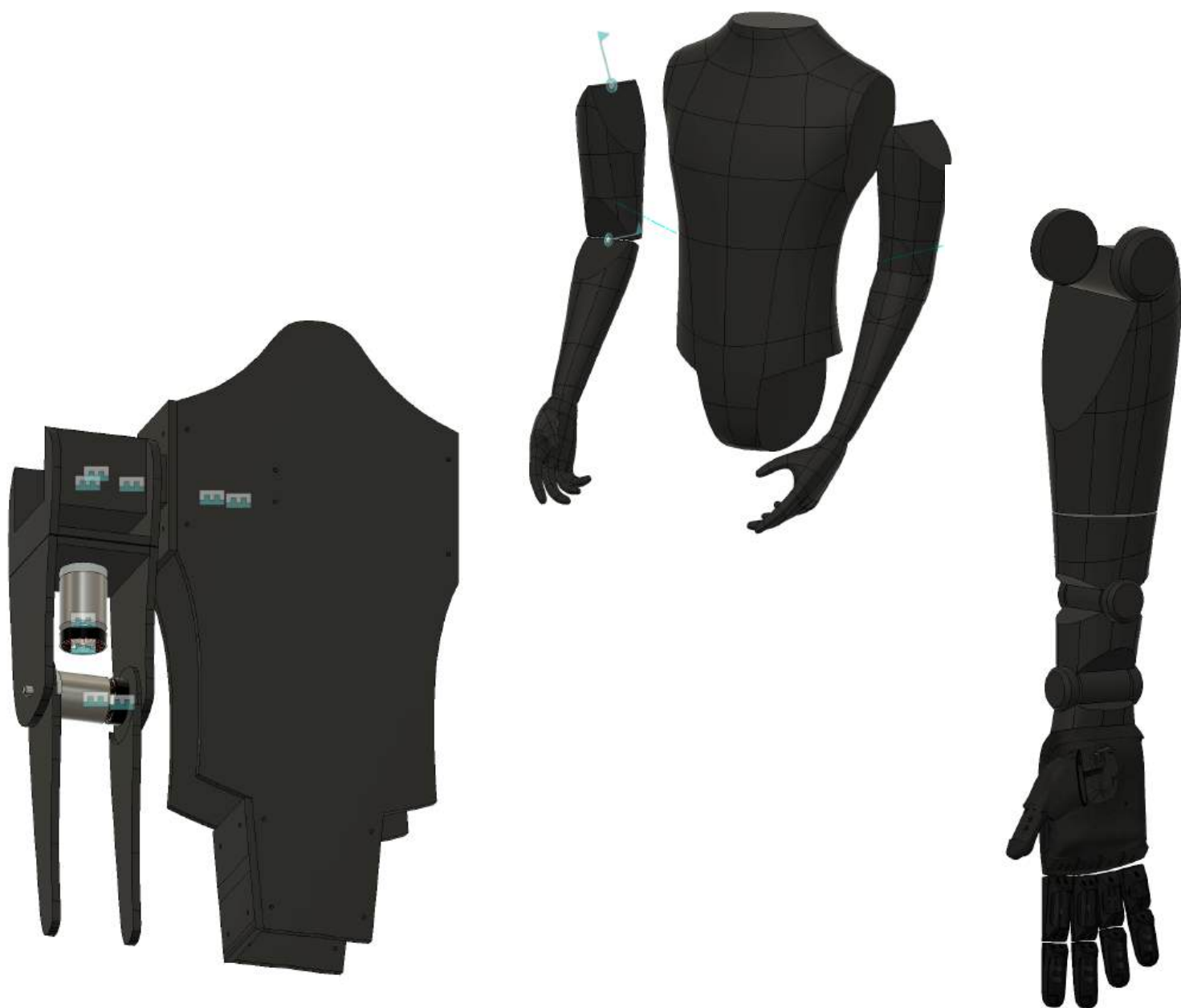


Humanoid > V2.3		
	<b>Bicep</b> 19/11/24 12:45 AM	V8 ▾
	<b>Face</b> 17/11/24 7:42 PM	V10 ▾
	<b>Forearm</b> 19/11/24 12:45 AM	V7 ▾
	<b>Full Bot Assembly</b> 24/11/24 2:50 PM	V7 ▾
	<b>Hand</b> 19/11/24 12:41 AM	V4 ▾
	<b>Head</b> 17/11/24 7:42 PM	V3 ▾
	<b>Torso</b> 23/10/24 11:07 AM	V5 ▾



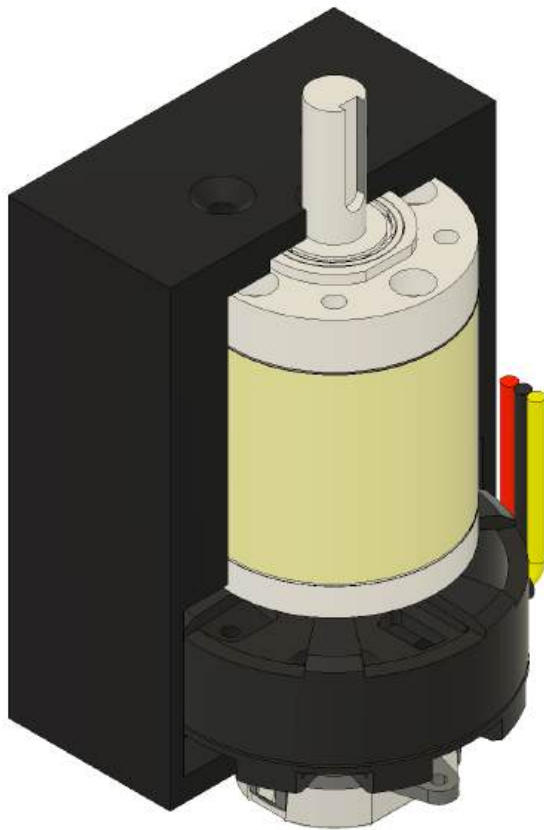
Humanoid	
	V1
	V1.1
	V2
	V2.1
	V2.2
	V2.3
	V3
	V3.1
	V4
	V4.1
	V5
	V7








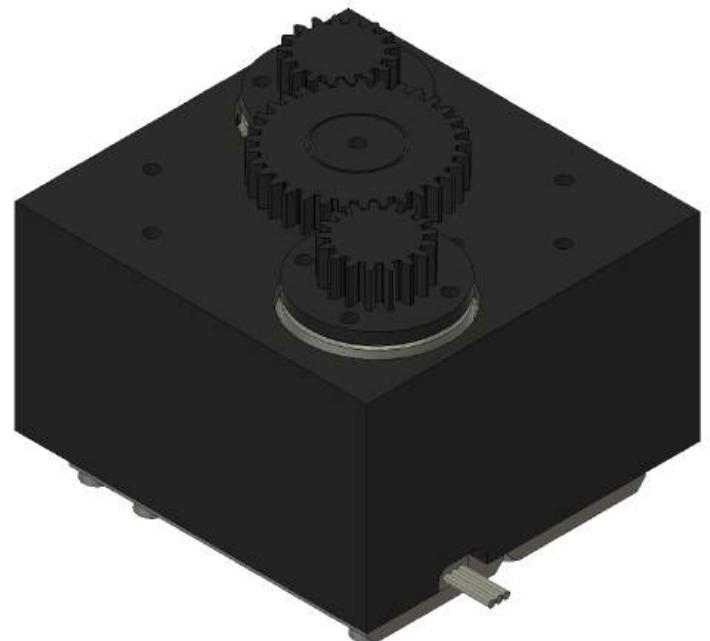
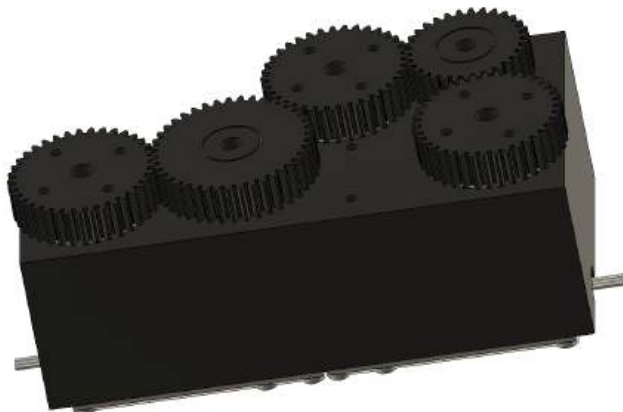


# Chapter 3: Actuator Designs

(February 2025)



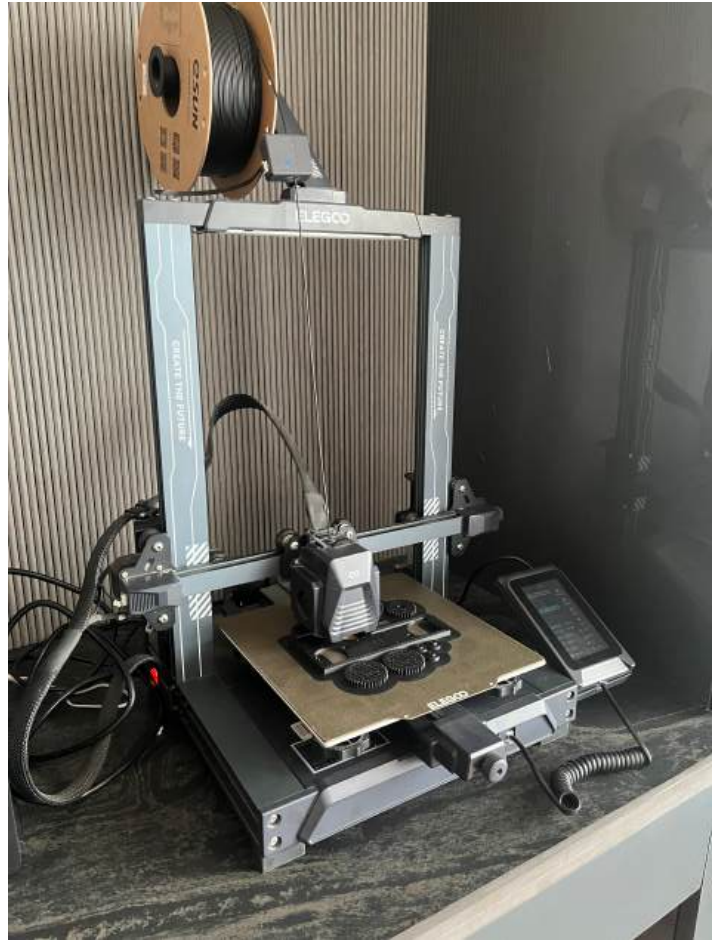
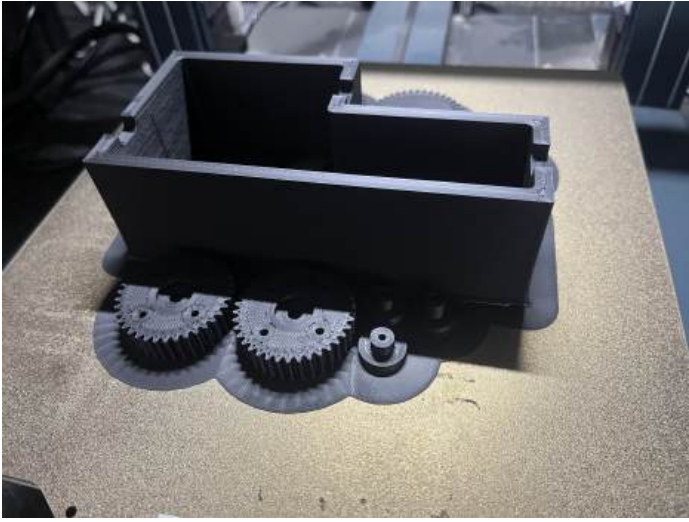
	<b>Servo Box Bicep</b> 18/02/25 1:18 PM V19 ▾
	<b>Servo Box Bicep v2 (timing belt)</b> 19/03/25 11:43 AM V6 ▾
	<b>Servo Box Bicep v3 (longer axes)</b> 21/03/25 1:29 PM V8 ▾
	<b>Servo Box Upper Bicep</b> 15/02/25 11:37 AM V6 ▾
	<b>Testing Gear</b> 15/02/25 6:35 PM V2 ▾





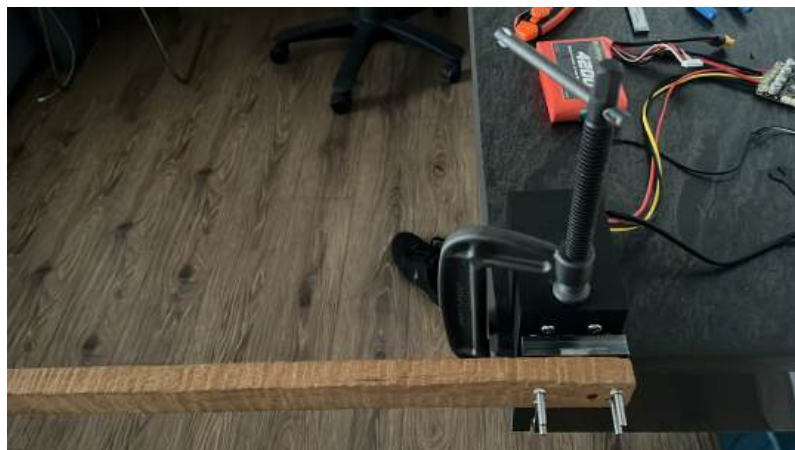
# Chapter 4: First Prints

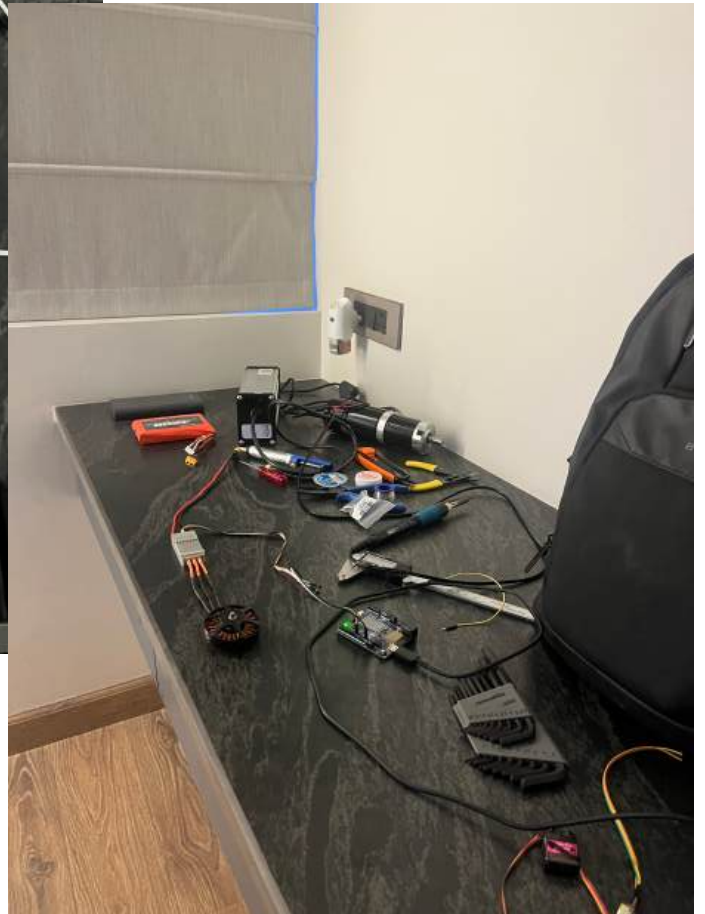
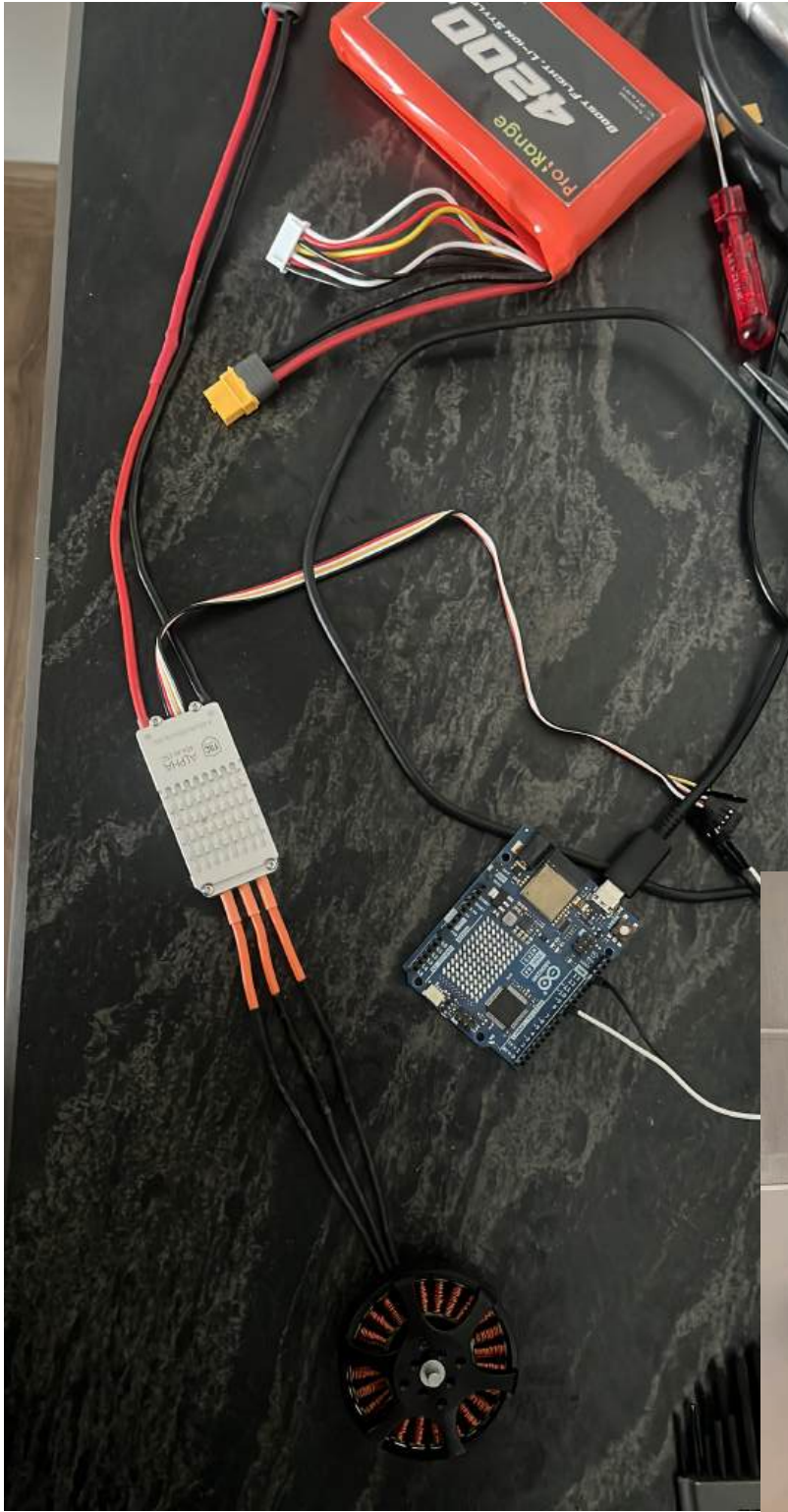
(March 2025)



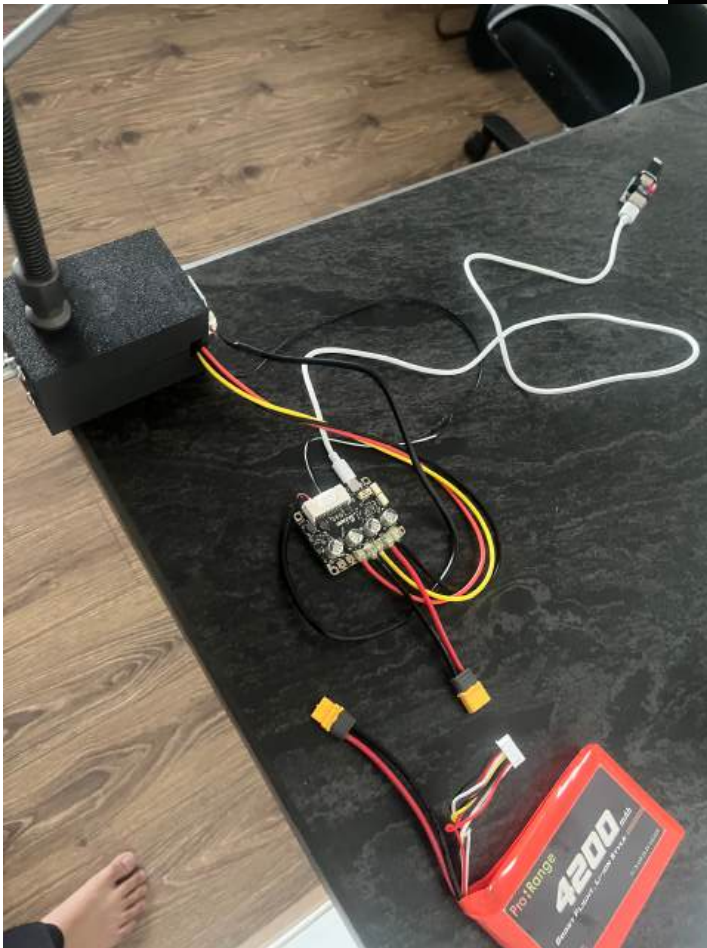
# Chapter 5: Testing (and failure)

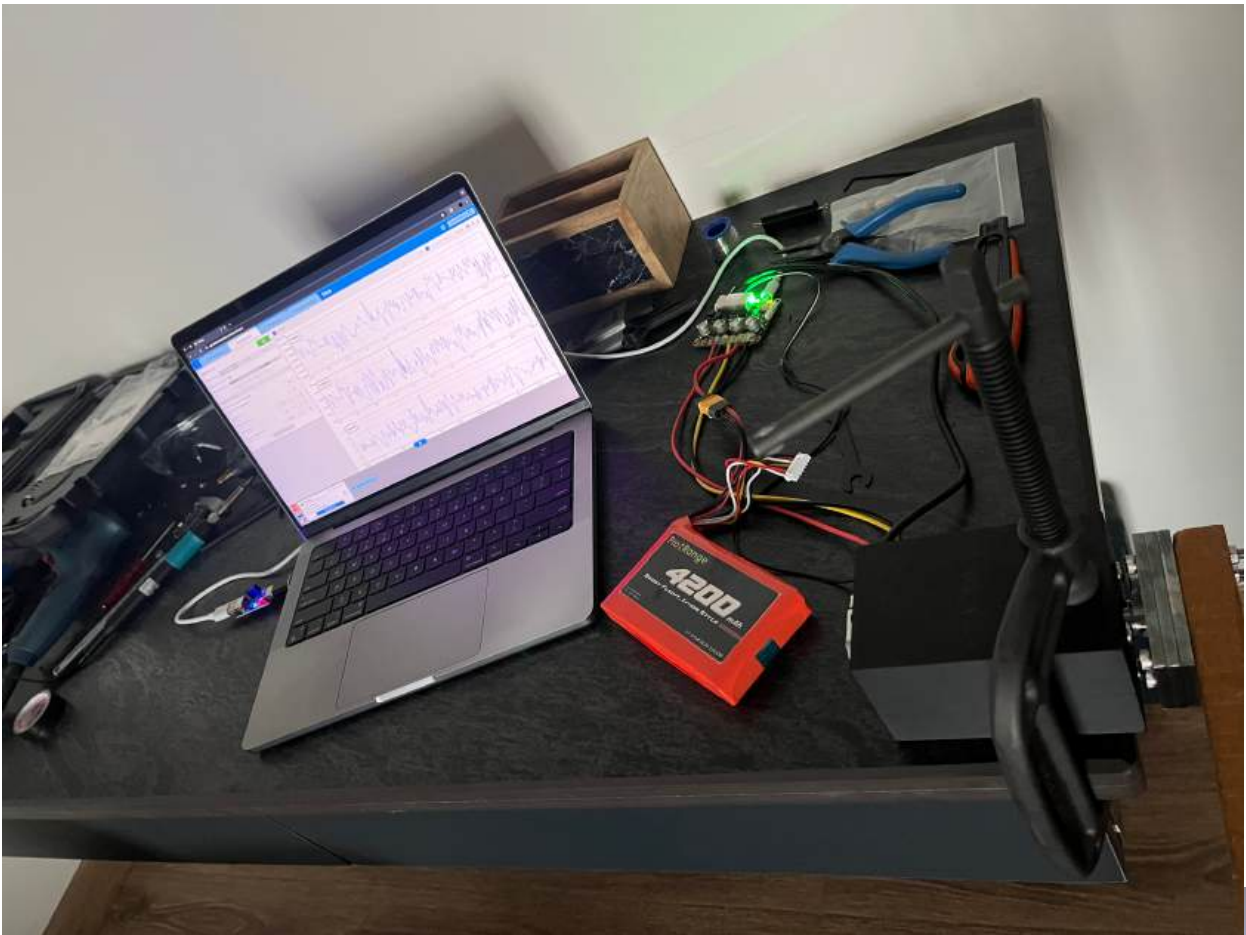
(March 2025)







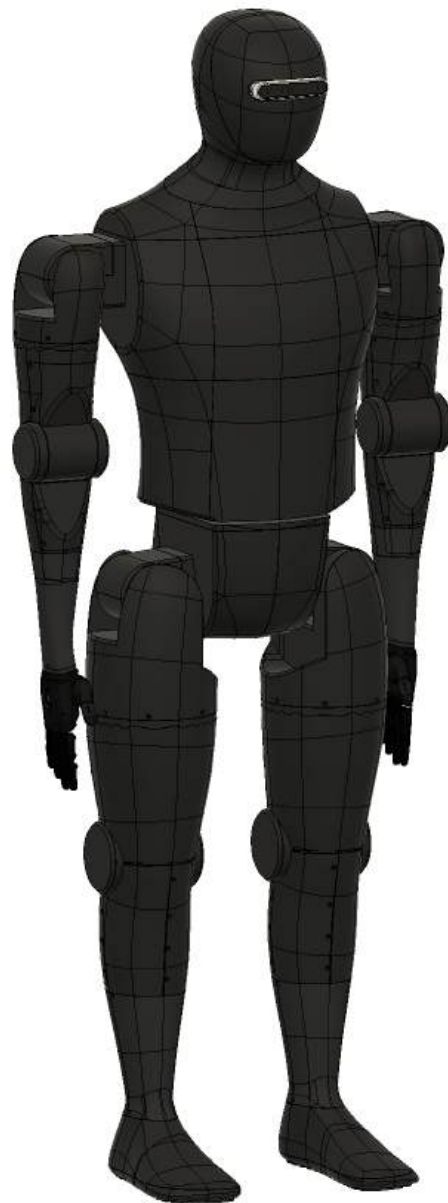


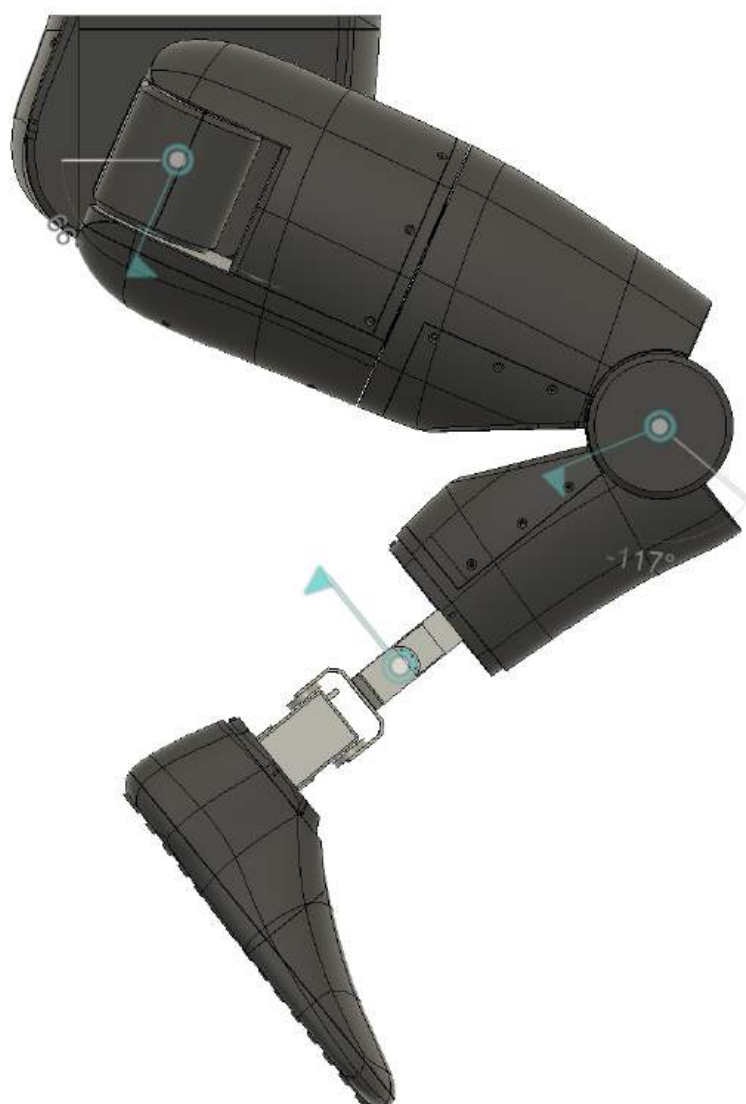
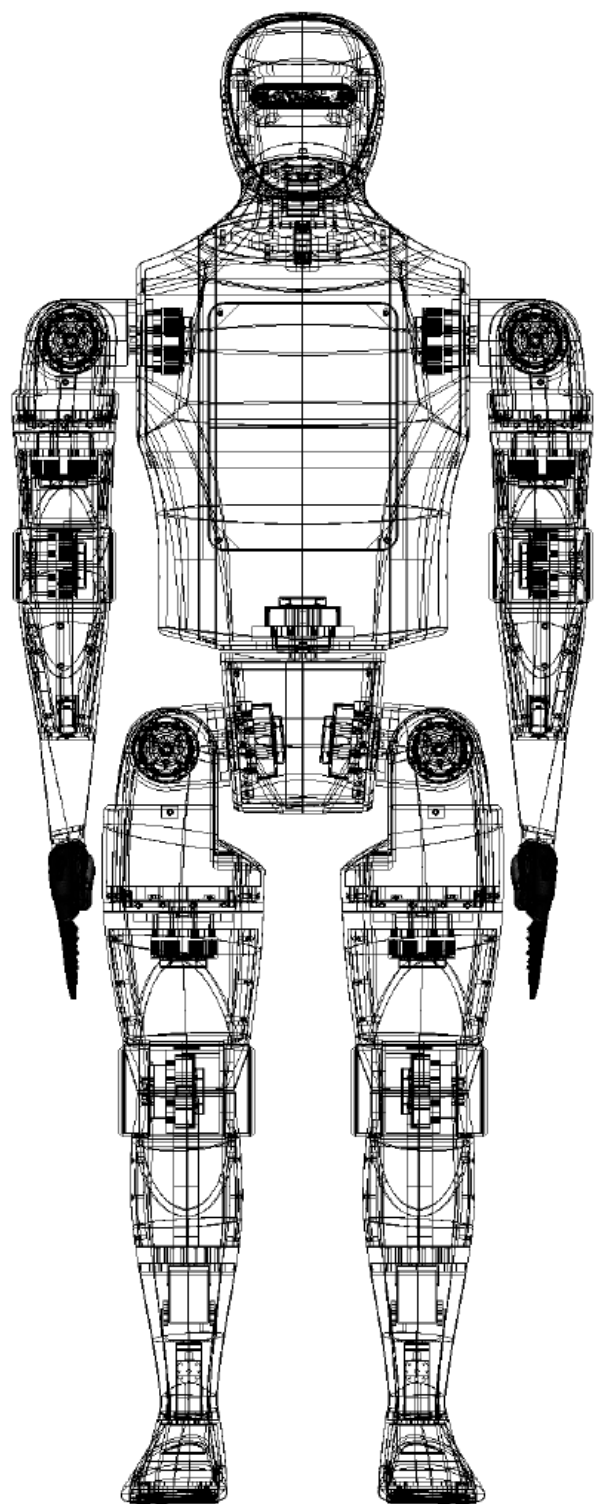




# Chapter 6: Back to the Drawing Board

(May 2025)





# Chapter 7: Building the First Prototype

(July 2025)











# Chapter 8: First Prototype Complete

(September 2025)

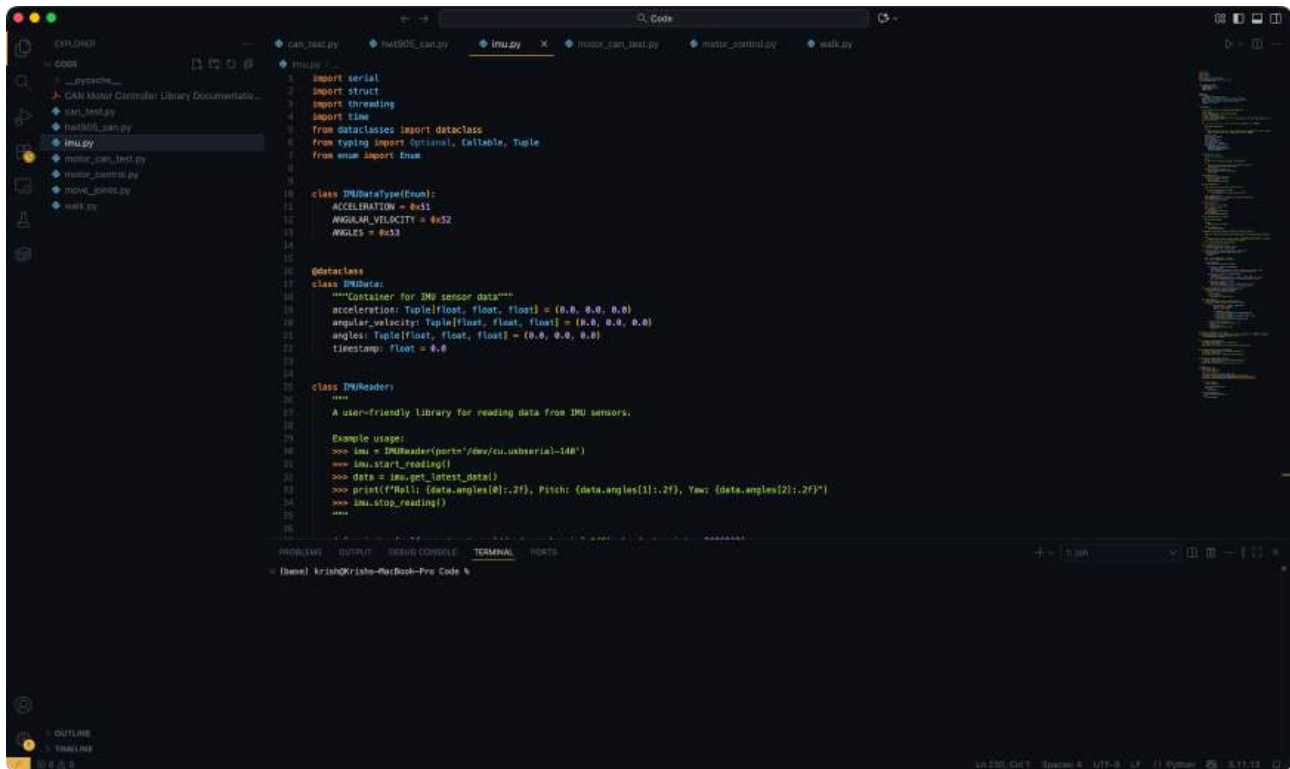




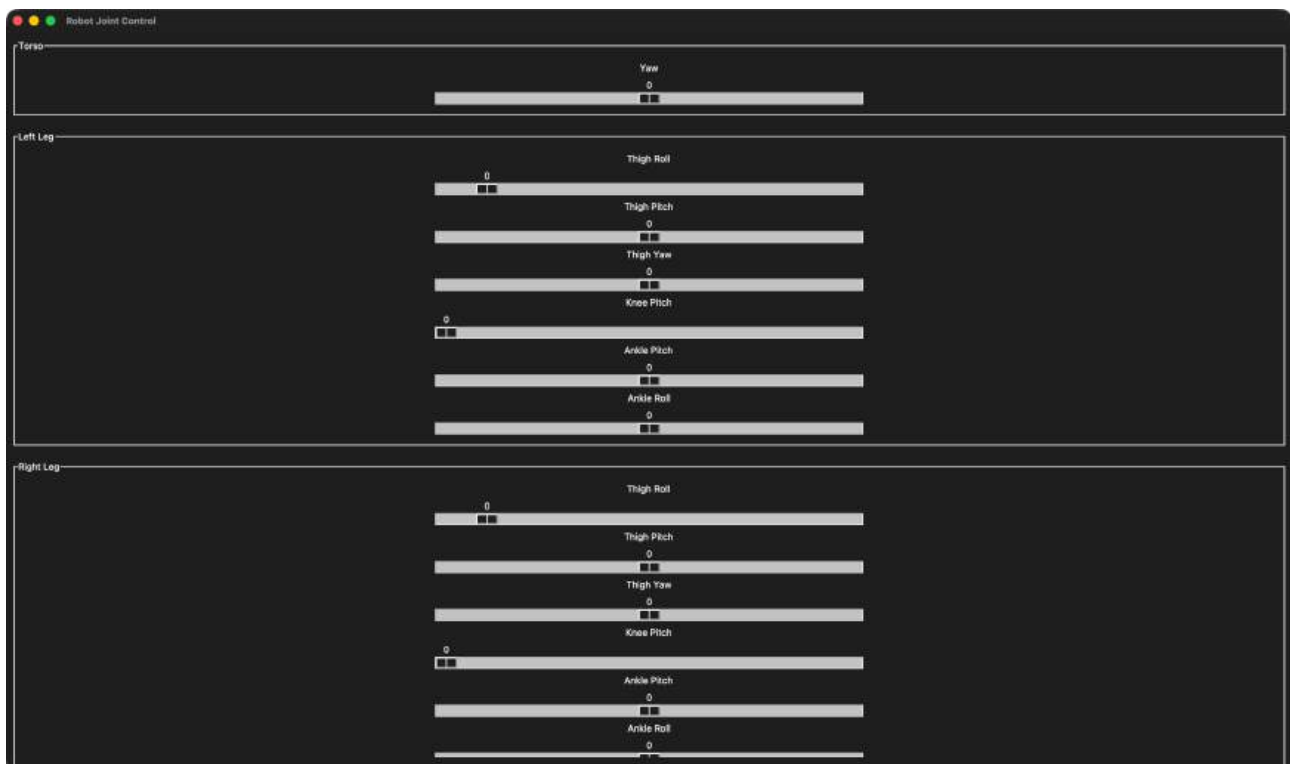


# Chapter 9: Programming

(October 2025)



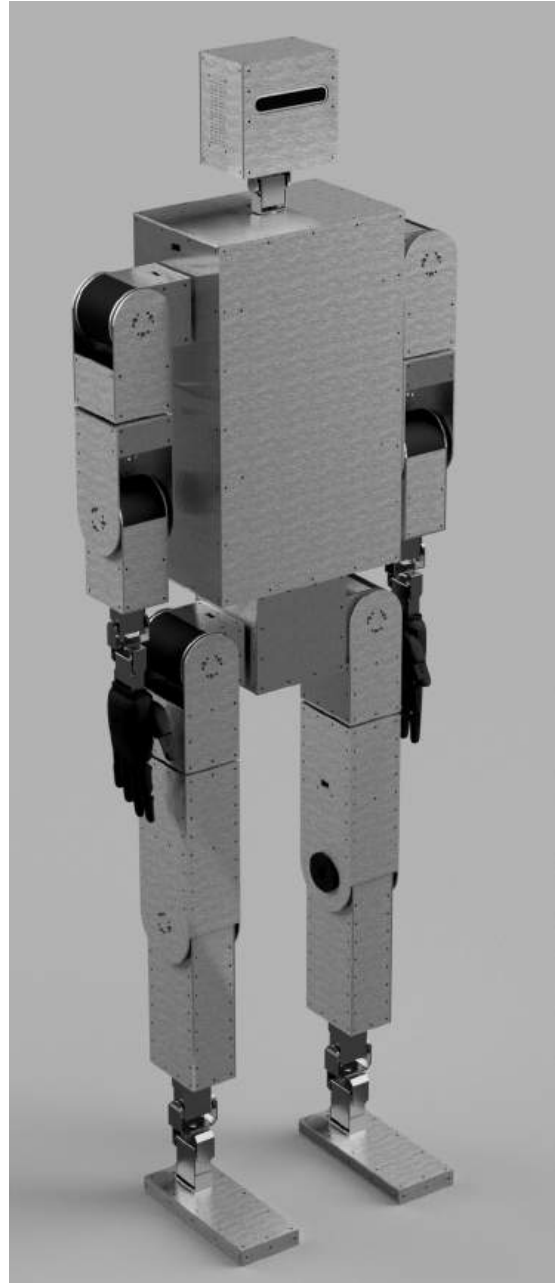
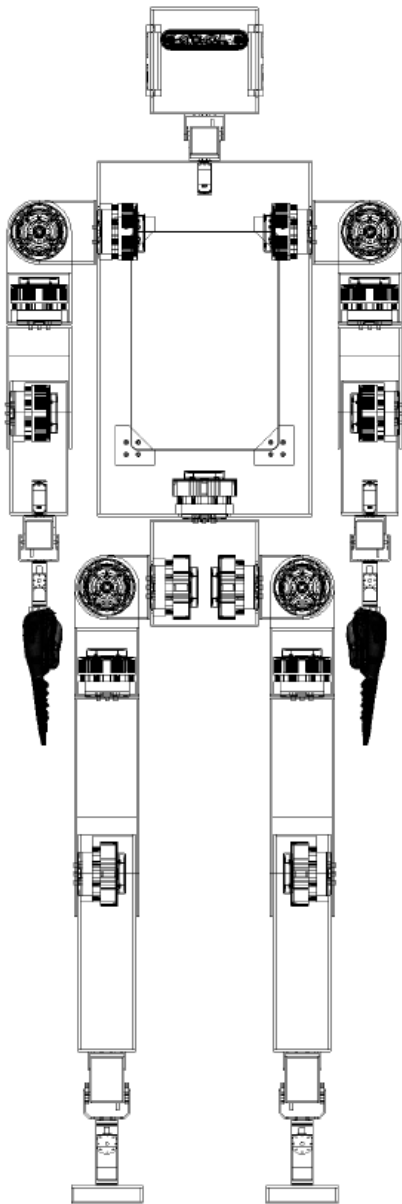
```
1 import serial
2 import struct
3 import threading
4 import time
5 from dataclasses import dataclass
6 from typing import Optional, Callable, Tuple
7 from enum import Enum
8
9
10 class IMUDataType(Enum):
11     """
12     Acceleration = 0x01
13     Angular Velocity = 0x02
14     Angles = 0x03
15 """
16
17 @dataclass
18 class IMUData:
19     """
20     Container for IMU sensor data"""
21     acceleration: Tuple[float, float, float] = (0.0, 0.0, 0.0)
22     angular_velocity: Tuple[float, float, float] = (0.0, 0.0, 0.0)
23     angles: Tuple[float, float, float] = (0.0, 0.0, 0.0)
24     timestamp: float = 0.0
25
26
27 class IMUReader:
28     """
29     A user-friendly library for reading data from IMU sensors.
30
31     Example usage:
32     >>> imu = IMUReader(port="/dev/cu.usbserial-14A")
33     >>> imu.start_reading()
34     >>> data = imu.get_latest_data()
35     >>> print(f"Roll: {data.angles[0]:.2f}, Pitch: {data.angles[1]:.2f}, Yaw: {data.angles[2]:.2f}")
36     >>> imu.stop_reading()
37     """
38
39     def __init__(self, port: str):
40         self.port = port
41         self.serial = serial.Serial(port, 115200, timeout=1)
42         self.data = IMUData()
43         self._reading_thread = None
44
45     def start_reading(self):
46         """Start reading data from the IMU sensor."""
47         self._reading_thread = threading.Thread(target=self._read_data)
48         self._reading_thread.start()
49
50     def _read_data(self):
51         """Read data from the IMU sensor in a loop."""
52         while True:
53             data = self._read_one_packet()
54             self.data = data
55
56     def _read_one_packet(self) -> IMUData:
57         """Read one packet of data from the IMU sensor.
58         Returns:
59             IMUData: The data read from the sensor.
60         """
61         # Read the header
62         header = self.serial.read(4)
63         if len(header) != 4:
64             return IMUData()
65
66         # Parse the header
67         accel_type, ang_vel_type, angles_type = struct.unpack('3B', header)
68
69         # Read the data
70         data = self.serial.read(24)
71         if len(data) != 24:
72             return IMUData()
73
74         # Parse the data
75         (acc_x, acc_y, acc_z, ang_vel_x, ang_vel_y, ang_vel_z,
76          angles_roll, angles_pitch, angles_yaw, timestamp) = struct.unpack(
77             f'{acc_type}f{ang_vel_type}f{angles_type}f', data)
78
79         # Return the data
80         return IMUData(
81             acceleration=(acc_x, acc_y, acc_z),
82             angular_velocity=(ang_vel_x, ang_vel_y, ang_vel_z),
83             angles=(angles_roll, angles_pitch, angles_yaw),
84             timestamp=timestamp
85         )
86
87     def stop_reading(self):
88         """Stop reading data from the IMU sensor.
89         Returns:
90             IMUData: The last data read from the sensor.
91         """
92         if self._reading_thread is not None:
93             self._reading_thread.join()
94         return self.data
95
96     def get_latest_data(self) -> IMUData:
97         """Get the latest data read from the IMU sensor.
98         Returns:
99             IMUData: The latest data read from the sensor.
100         """
101         return self.data
```

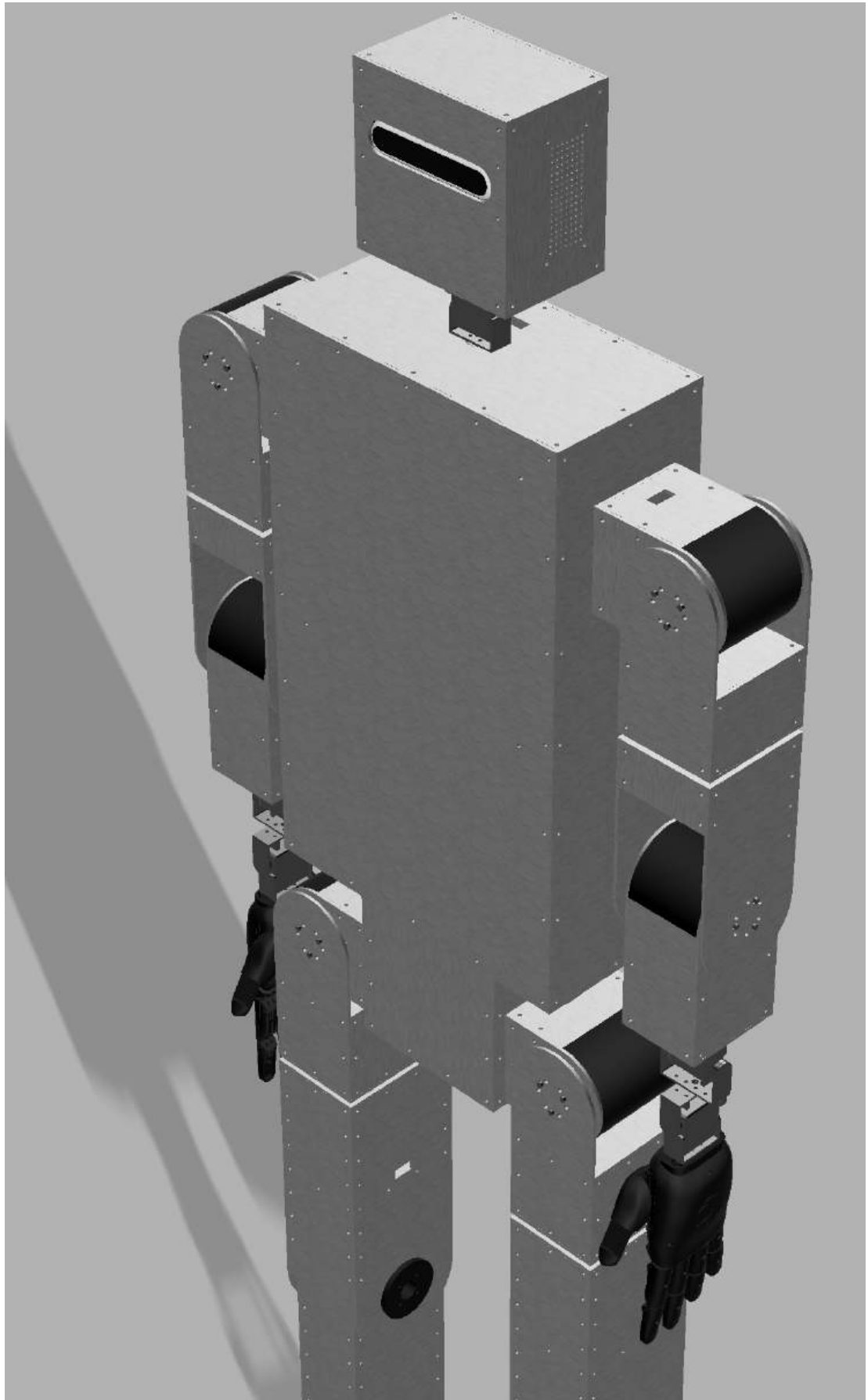




# Chapter 10: Designing the Second Prototype

(October 2025)





# Chapter 11: Building the Second Prototype

(Ongoing)







