Glove for Real-time Imitation and Performance

Group 21:
Rian Lowery
Kane McAgy
Christopher Marcoccia
Griseld Xhengo

Team Members

Rian Lowery
Computer Engineering

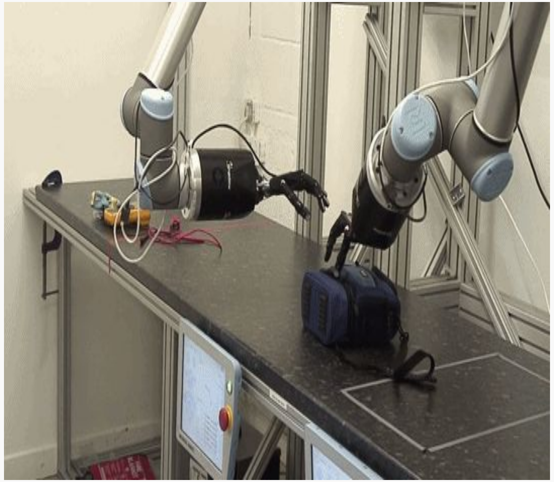
Kane McAgy
Electrical Engineering

Christopher Marcoccia
Electrical Engineering

Griseld Xhengo
Electrical Engineering

Project Problem, Motivation, and Applications

- Some environments and materials are too hazardous for direct human contact
- Current remote interaction methods are limiting
- GRIP can maintain natural interactions without risk to personnel



Experimentation

Technological interaction with exterior environments and harmful substances can mitigate necessity for human contact participation in experimentation.

Exploration

-Attempts to give remote operators more direct interactions with the environment they are studying.
-Remotely piloted submersibles with controllable appendages.

Goals and Objectives

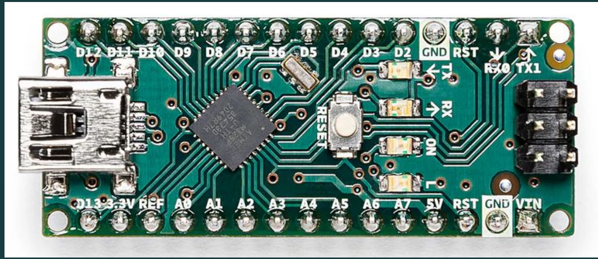
Goal type	Description
Basic	• Create a device with a series of sensors that will track the user's hand movements and translate them to GRIP, allowing it to mimic the user with high accuracy. • The hand will be able to have an open or closed grip as well as other standard positions
Advanced	• The system will be able to sustain a low-latency connection with moderate traffic between the user's hand and GRIP, allowing the hand to move with the user with negligible delay. • The hand portion of GRIP will have articulated fingers that can move independently of each other, Sallowing for more intricate interactions with the environment. Basic movements such as pointing and pinching will be doable.
Stretch	• Incorporate an arm attached to the hand that will be able to mimic the user's arm movement. • Single axis wrist movement • A homing algorithm will be made to calibrate soft and hard stops for the arm, allowing use from different-sized users after the calibration sequence.

Engineering Specifications

Category	Parameter	Requirement / Component
Physical	Total Mass	≤ 2.3kg/Hand, ≤ 0.5kg/Glove
	Fit	One size
Performance	Runtime	≥ 2 hours
	Latency	≤ 200 ms
Communication	Link Reliability	Packet loss ≤ 0.5% at 5 m LOS
	Wireless Transmitter	nRF24L01 module
Control	Microcontroller	Arduino Nano
Actuation / Interface	Finger Sensor	ZD10-100

MCU Selection

	Arduino Nano	Teensy 4.1	ESP32
Programing Method	Arduino IDE	C/C++ in Arduino IDE	C/C++ in Arduino IDE
Operating Voltage	5V	3.3V	3.3V
Size	18 x 45 mm	17.78 x 61.66 mm	28 x 55.30 mm
Cost	$25.70	$31.50	$11

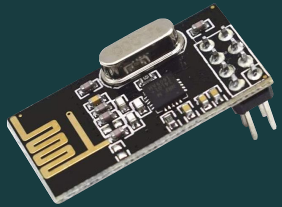


Motor Selection

	Servo Motor (MG996R)	Stepper Motor
Control	Closed loop	Open loop
Position Accuracy	High	Can lose steps under load
Return to Position	Automatic	Requires a command
Ease of Control	Simple signal	Step and direction
Holding Position Torque	Consumes Power	Excellent

Wireless Communication

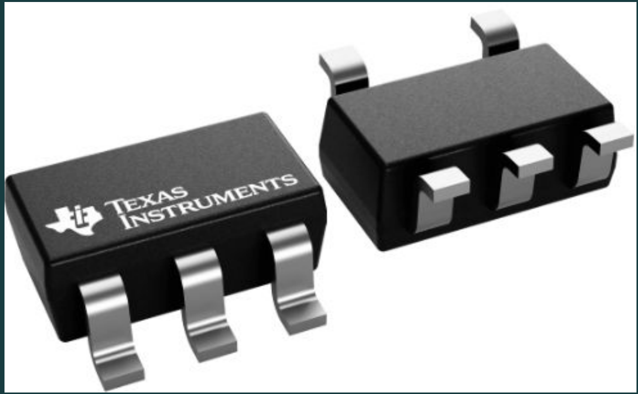
Part	RFM69	nRF24l01
Frequency	315-915 MHz	2.4 GHz
Range	~300m	~50-100m
Penetration	High	Low
Bit Rate	300kbps	2Mbps
Cost	~$9.00	~$2.00



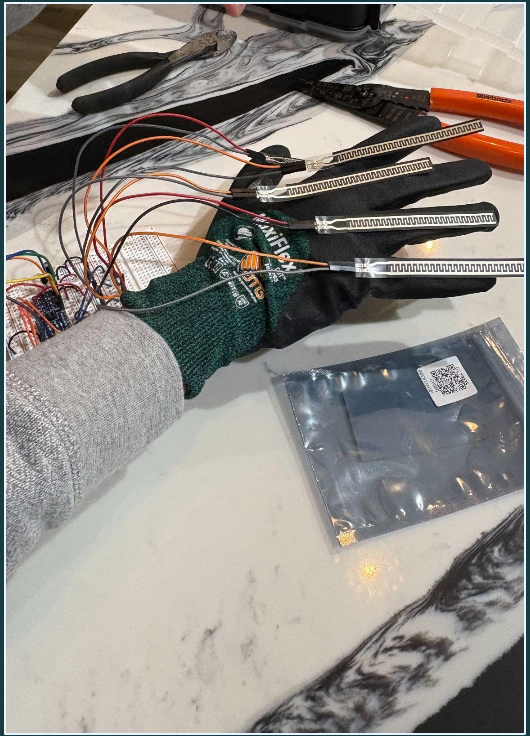
- While more volatile than the RFM, the nRF has significantly faster data transmission speeds
- Range and penetration can be improved upon easier than packet transmission

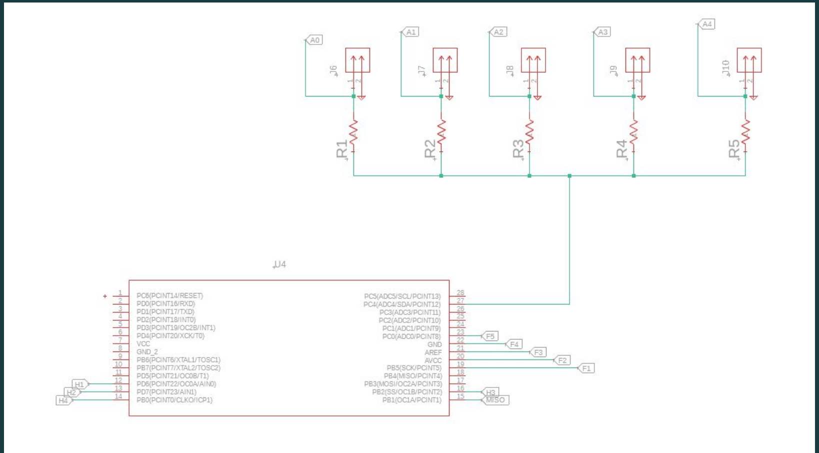
Voltage Regulator

	LM3670	TLV62569
Max Iout	0.35A	2A
Vout Range	0.7V - 3.3V	0.6V - 5.5V
Vin Range	2.5V - 5.5V	2.5V - 5.5V
Operating Temperature (°C)	-40 - 85	-40 - 125



Flex Sensors - Voltage Divider Circuit

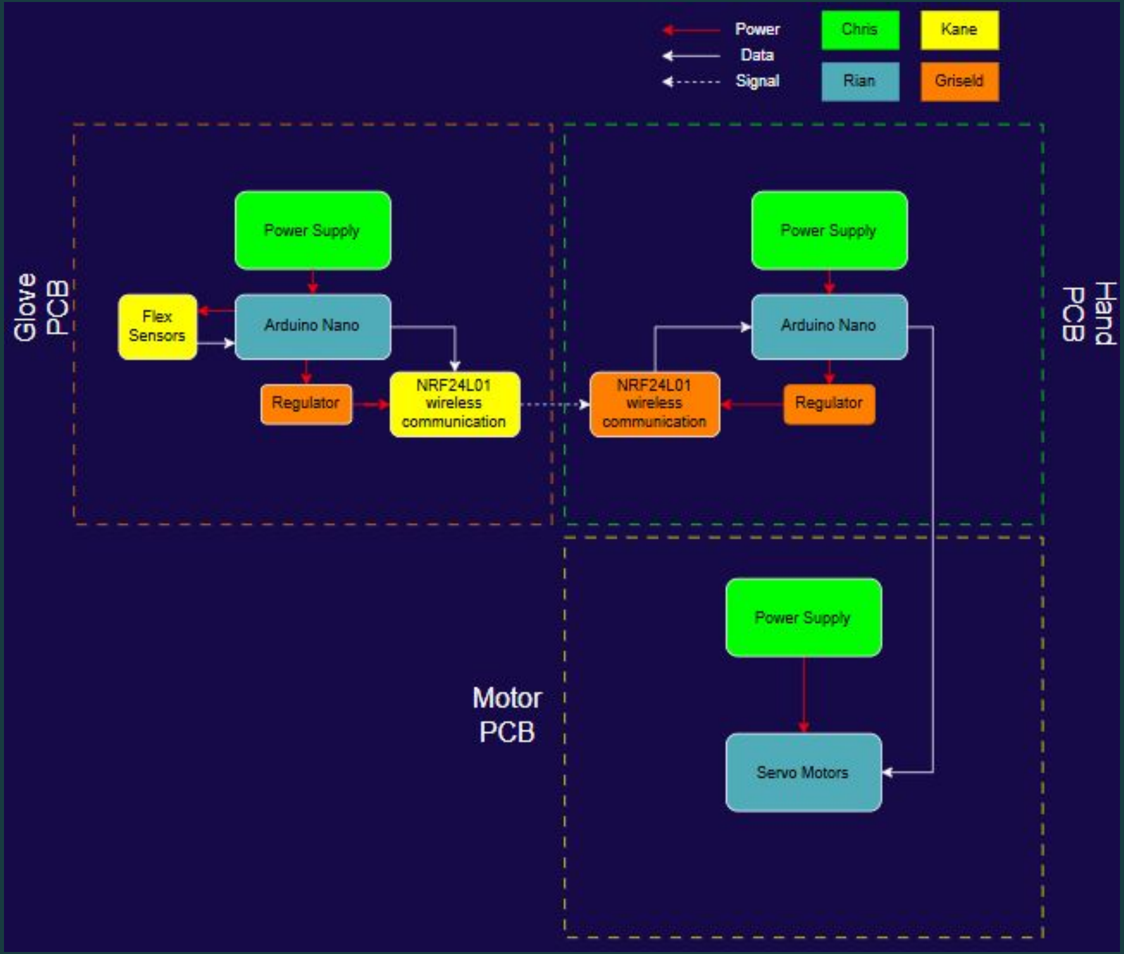




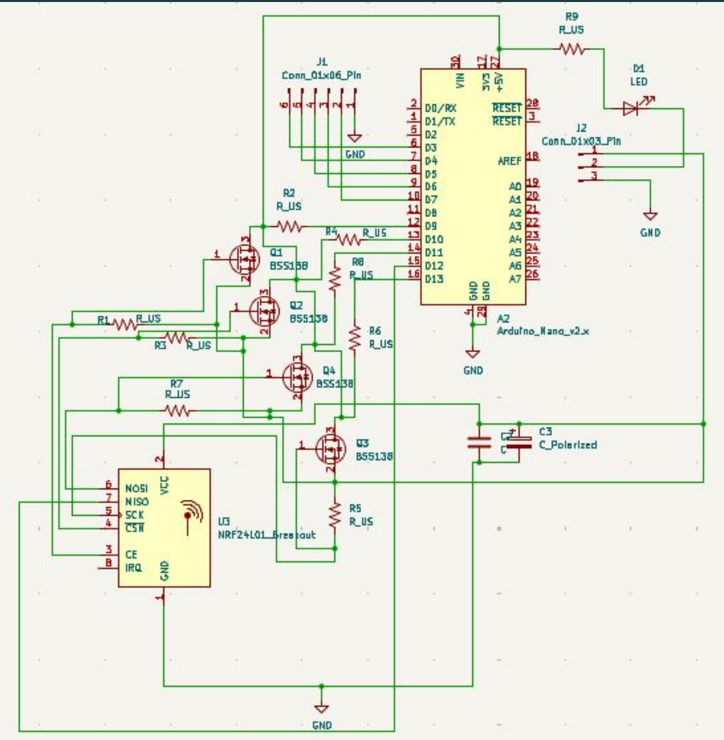
Voltage Divider Circuit to PCB connections

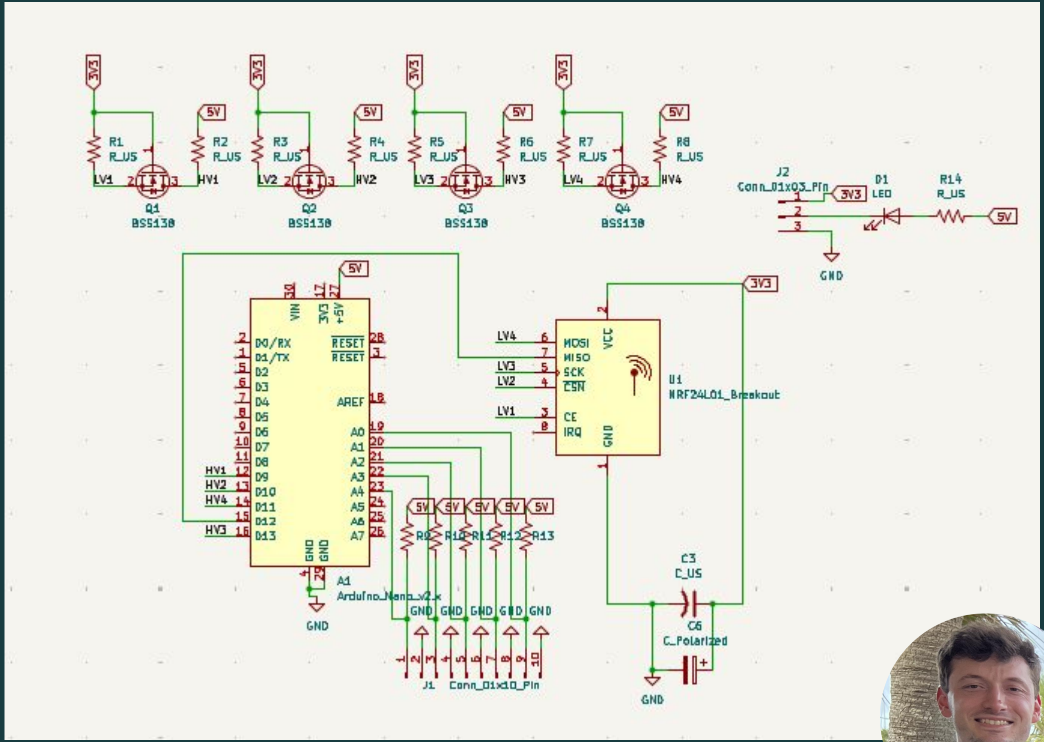
- Flex sensor signals enter the Arduino Nano’s ADC ports
- Sensor’s output ~0–1023 (10-bit)
- Baseline calibration normalizes finger bend differences
- Smoothing via moving average / low-pass filtering
- Convert mapped value to joint-angle range (0–180°)

Hardware Block Diagram

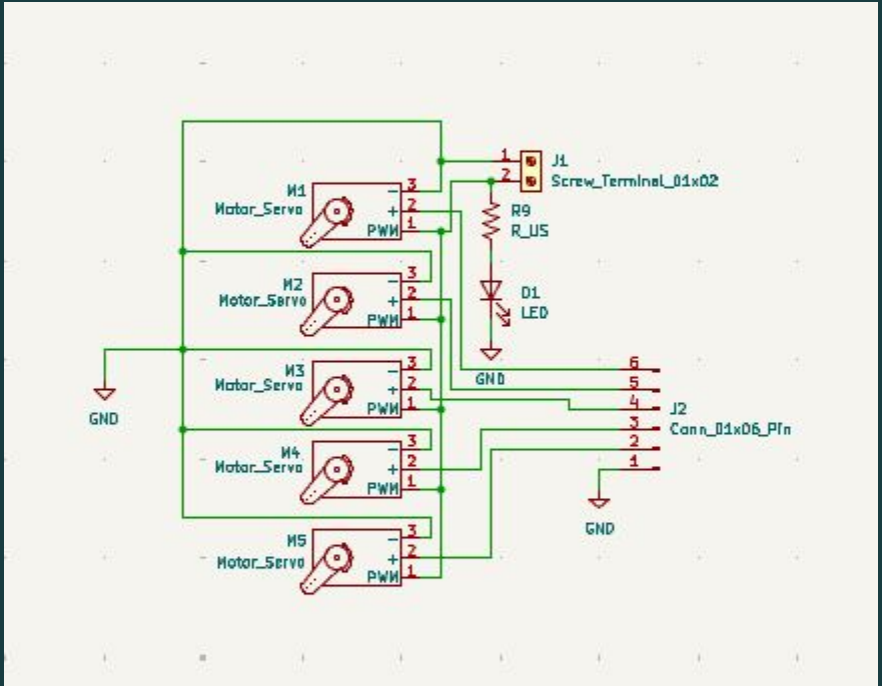


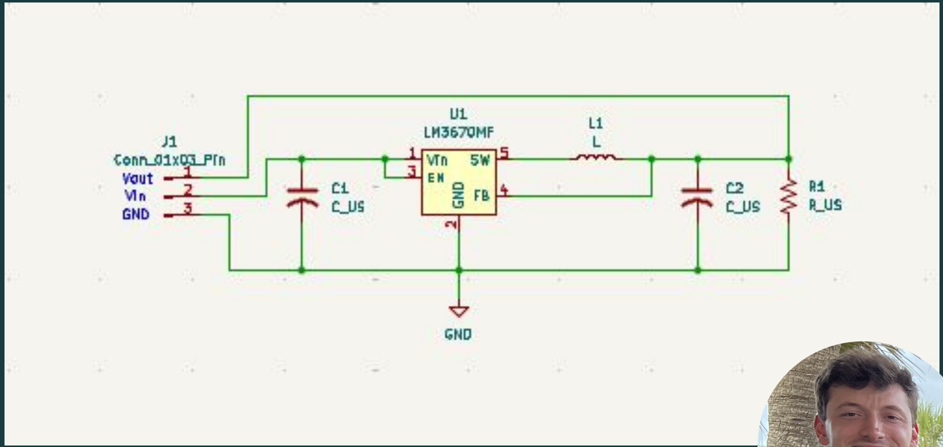
Schematic for the Hand and the Glove





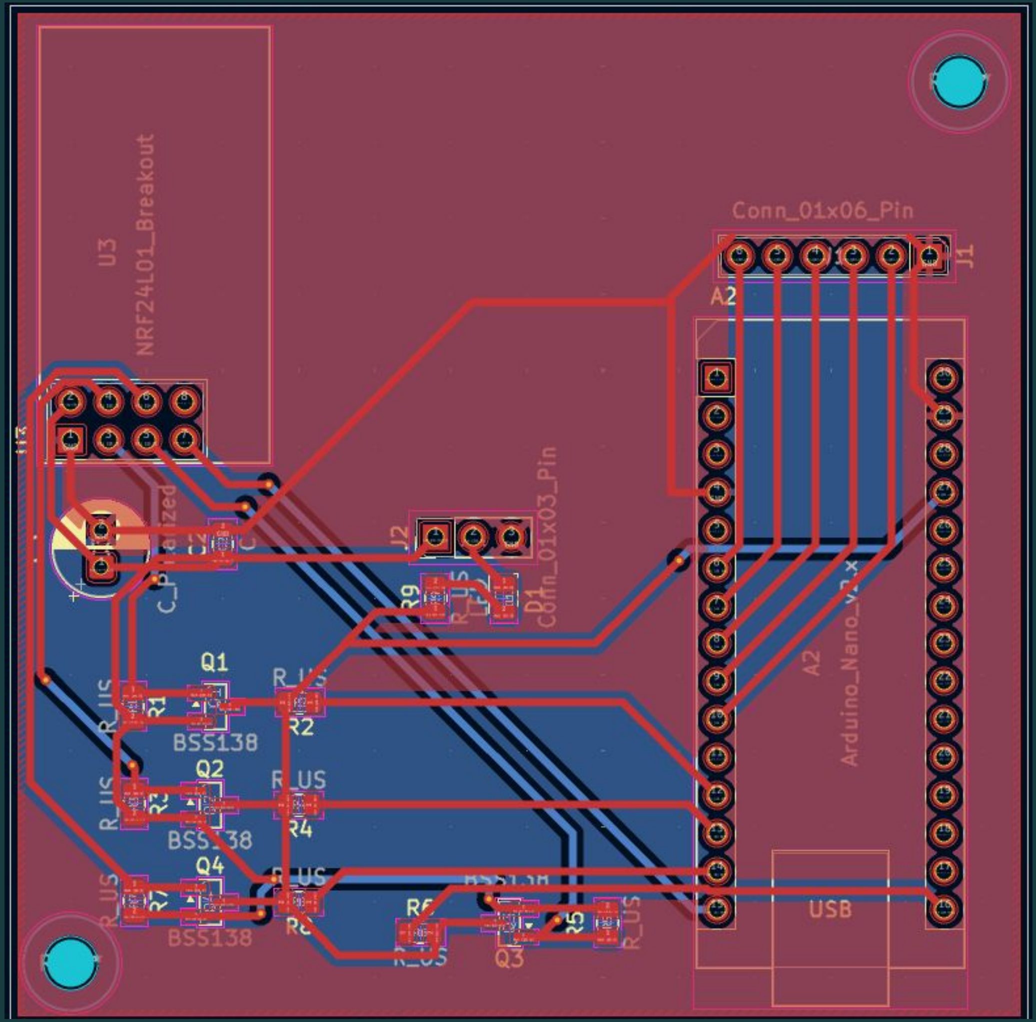
Schematics for the Motors and the Regulator





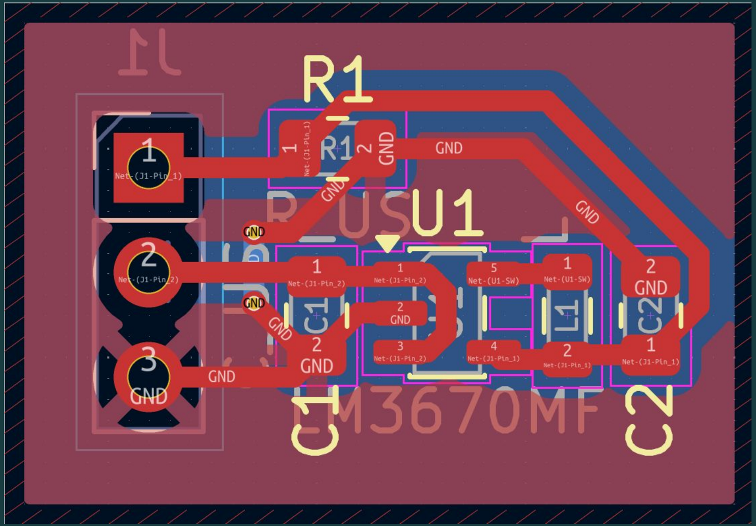
PCB Design for the Hand

The wireless communication will receive data from the unit on the glove and the arduino will convert the values to be usable by the motors on the servo board connected to the pins on this board.

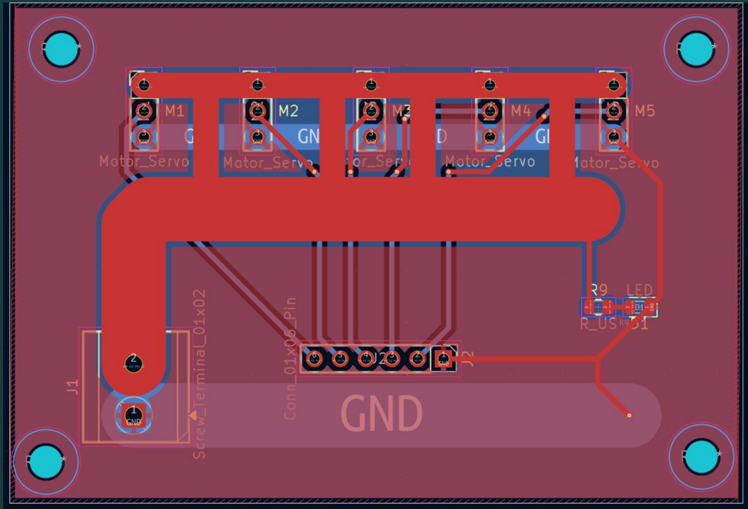


PCB Design for the Regulator and the Motors

The regulator steps the voltage down to 3.3V and 0.3A which will be more than enough for the wireless communication module on both the hand and the glove PCBs.



The PCB for the servos is separate because of the high voltage and current that they will require. It will use a separate power source just for the motors and connect to the hand PCB via connectors and wires.



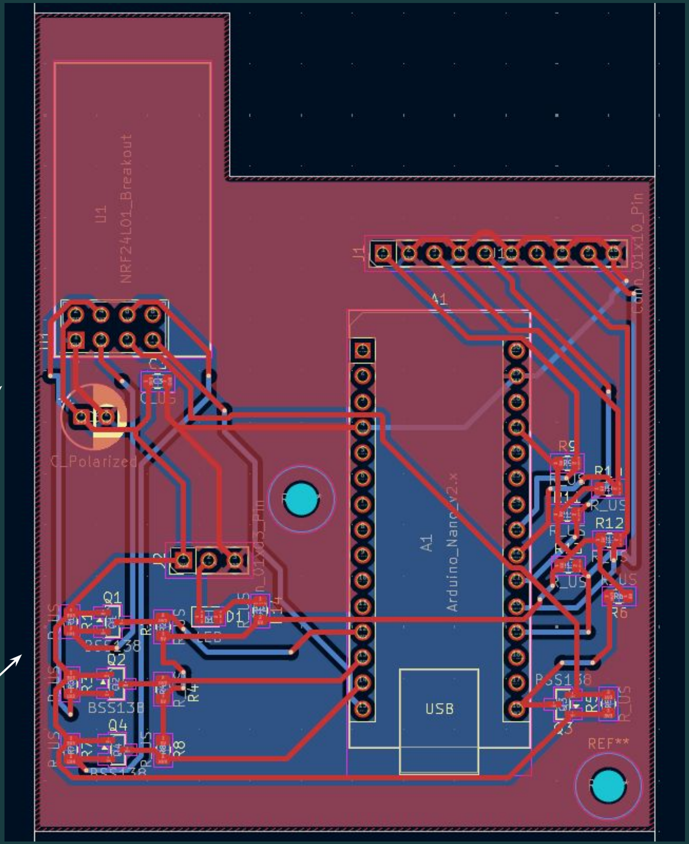
Glove PCB Design

Five flex sensors that will take the users movements and convert them into values that will reflect the hand movements.

The data is transmitted by the wireless transmitter to the receiver on the hand PCB.

Transceiver

Enables Wireless communication



Voltage Divider Flex Sensor feed (Fingers)

Software Architecture Overview

Glove

Sensor acquisition, calibration, filtering, and angle mapping

- Analog flex sensor acquisition
- Signal conditioning (normalization + curve shaping)
- Joint angle generation

01

Wireless

Low-latency packet transmission via nRF24L01

- Structured packet formation
- Hardware CRC validation
- 50–200 Hz update rate

02

Receiver

Packet decoding and safety validation

- Packet parsing
- Angle extraction
- Timeout detection (failsafe)

03

Servos

Real-time actuation of robotic joints

- Servo angle mapping
- Constrained motion limits
- Smooth positional updates

04

Design decisions supporting real-time performance

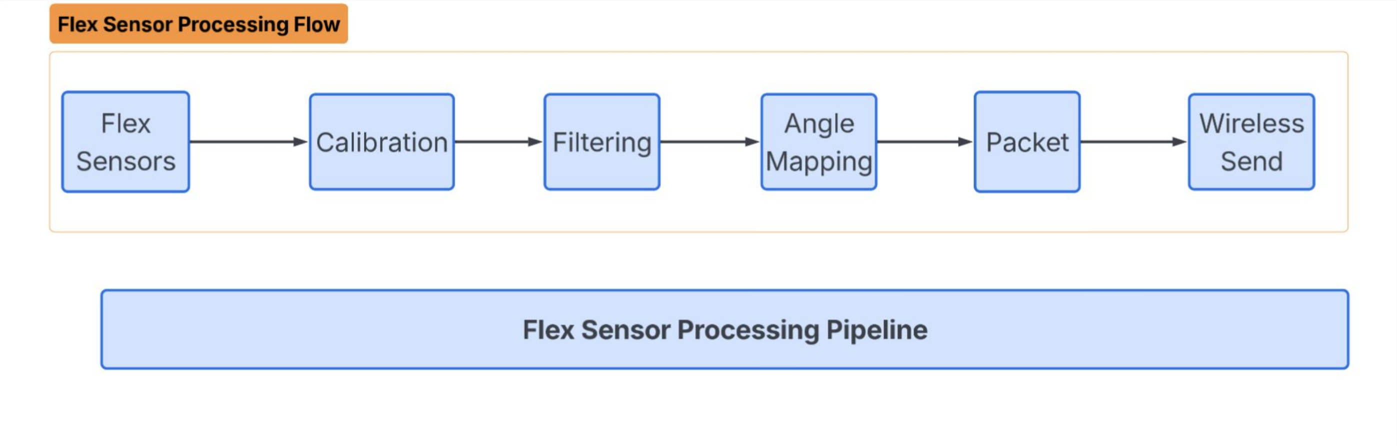
Software Technology Selection & Justification

Component	Options Considered	Selected	Justification
Programming Language	Python, C++	C++	Deterministic timing, low-level hardware access
Wireless Protocol	Bluetooth, WiFi, nRF24	nRF24	Low latency, hardware CRC, simple packet control
Control Architecture	Interrupt-based, Polling Loop	Polling Loop	Predictable timing and simpler synchronization
Servo Control Method	Software PWM, Hardware Timer	Servo Library (Hardware Timer)	Stable pulse timing with minimal overhead

Glove-Side Firmware Architecture

From finger bend → clean angle data → wireless packets (fast + smooth)

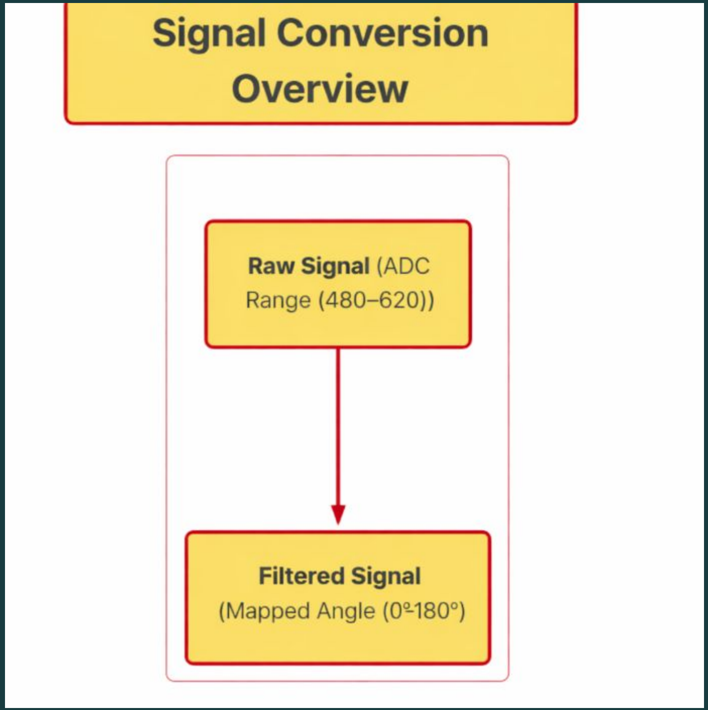
- Arduino Nano (Glove MCU)
- Analog Flex Sensor Acquisition
- User Calibration (Min/Max Mapping)
- Low-Pass Filtering (Noise Reduction)
- Angle Mapping (Sensor → Joint Angle)
- Structured Packet Formation (CRC Included)
- Wireless Transmission via nRF24L01 (200 Hz)



Calibration and Signal Conditioning

Signal conditioning pipeline ensuring stable, user-adaptive, low-latency motion tracking prior to wireless transmission

- Flex sensors vary per user
- Record min/max bend values during calibration
- Normalize sensor range (ADC → 0–100%)
- Apply low-pass filter to reduce noise
- Balance smoothness vs latency



Wireless Communication & Packet Structure

Low-latency, synchronized multi-finger transmission with hardware-level error detection

- nRF24L01 2.4 GHz transceiver
- 250 kbps data rate (stable mode)
- Custom multi-finger packet structure
- Built-in hardware CRC validation
- ~50 Hz update rate (scalable)

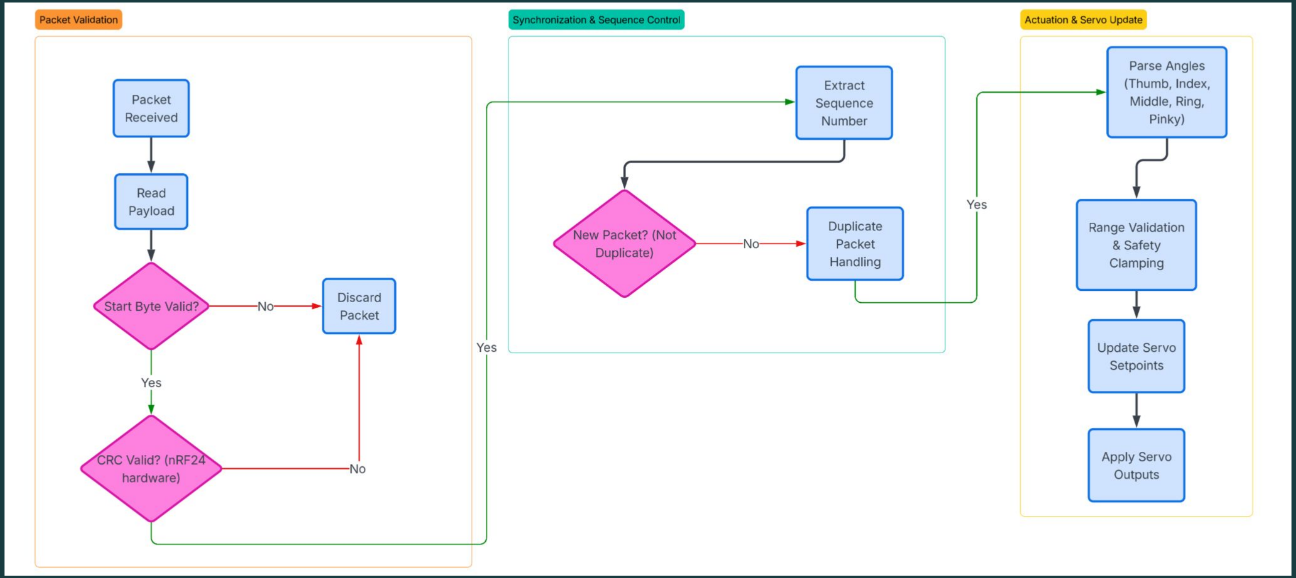
Wireless Data Packet (Glove → Receiver)

Start Byte (1 byte)	Sequence # (1 byte)	Thumb Angle (1 byte)	Index Angle (1 byte)	Middle Angle (1 byte)	Ring Angle (1 byte)	Pinky Angle (1 byte)	CRC (Hardware – nRF24)

Total Payload Size: 7 Bytes
CRC: Automatically handled by nRF24 hardware

Ensures synchronized, error-free angle reconstruction prior to servo actuation

Receiver-Side Processing & Validation



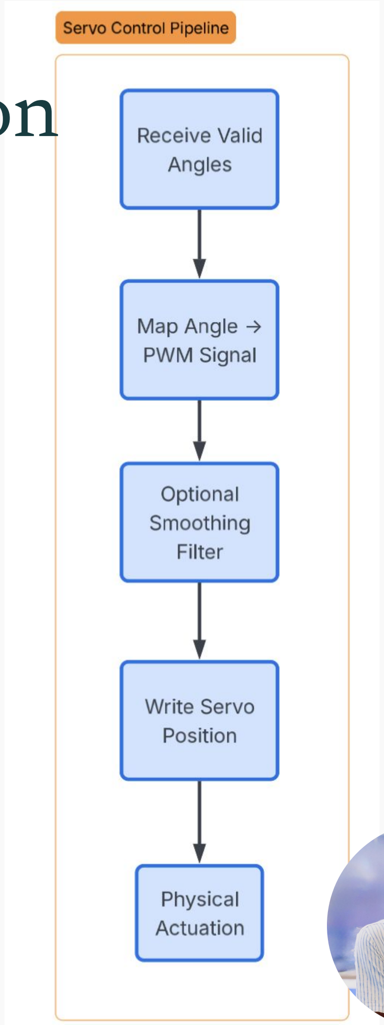
- Hardware CRC prevents corrupted data from reaching control logic
- Sequence tracking detects packet loss/duplicates
- Safety clamping enforces valid servo command bounds
- Failsafe: hold last valid position if packet rejected

Design Goal: Prevent corrupted, duplicated, or misaligned packets from destabilizing actuator motion

Servo Control & Real-Time Actuation

Deterministic real-time angle-to-actuator conversion with stability safeguards

- Map received joint angles to servo PWM signals
- 0–180° angle → calibrated pulse width output
- Real-time update at ~50 Hz
- Motion smoothing to reduce jitter
- Failsafe: hold last valid position on packet loss



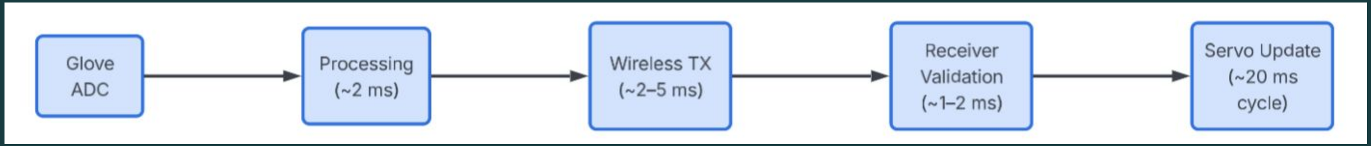
Performance Evaluation & System Metrics

Measured latency, update rate, and motion stability validation

Metric	Target Specification	Measured Result	Status
End-to-End Latency	< 15 ms	~12–18 ms (servo cycle limited)	Within Design Margin
System Update Rate	≥ 50 Hz	~50 Hz (20 ms loop)	Met
Wireless Data Integrity	CRC Validation Required	100% corrupted packets rejected	Met
Motion Stability	No visible jitter	Stable after tuning filter	Met
Packet Loss Handling	No unstable motion	Holds last valid position	Met

End-to-End Latency Breakdown

System meets real-time motion requirements while maintaining packet integrity and actuation stability



Design Challenges & Engineering Tradeoffs

Balancing latency, stability, reliability, and scalability

Technical Challenges

- Sensor noise and ADC variability
- Wireless packet timing unpredictability
- Servo PWM refresh cycle limiting latency
- Multi-finger synchronization
- Preventing unstable actuation on packet loss

Key Tradeoffs

- Filtering vs Responsiveness
- Data Rate vs Transmission Stability
- Update Frequency vs Servo Jitter
- Simplicity vs Protocol Robustness
- Immediate Reset vs Hold-Last-Position Safety

Design Objective: Maintain smooth, synchronized multi-finger motion under real-time constraints.

Opportunities for enhanced performance and robustness

Improvements & System Scalability

- *Increase servo update frequency with higher-speed digital servos*
- *Add interpolation for smoother multi-finger motion*
- *Expand packet structure for additional sensors*
- *Implement sequence numbering for advanced packet ordering*
- *Implement adaptive filtering based on motion speed*

Mounting and Design Breakdown

All enclosed:

- Glove capable of withstanding ~5V charge
- High dexterity was big factor for optimal performance

- Glove PCB

- Flex Sensors

- Velcro Arm band for Battery pack

Glove Assembly

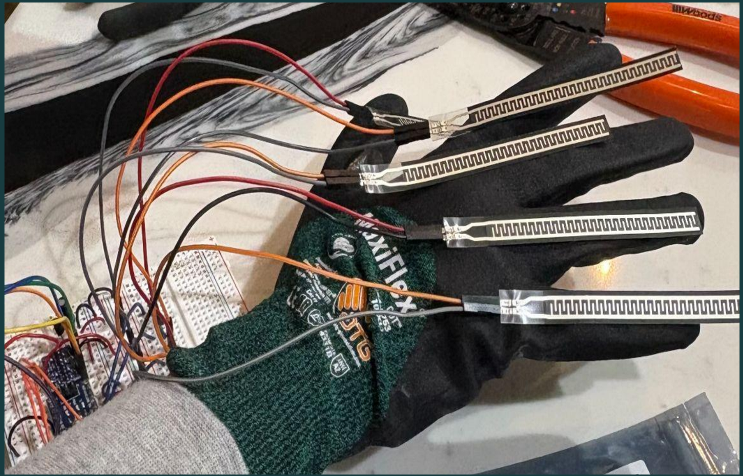
Flex Sensors - ZD10 - 100

The sensor was slowly bent from its minimum to maximum curvature while its analog output was monitored through the Arduino serial monitor.

Typical values recorded:

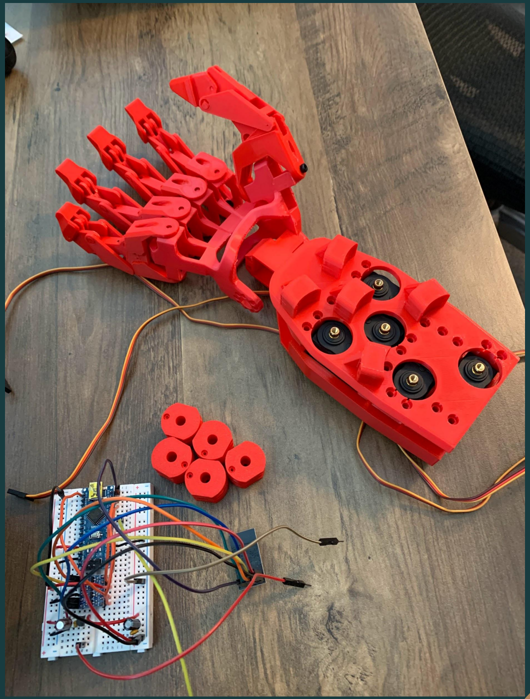
* **Unbent state:** ~980–1096
* **Half bend:** ~200–980
* **Full bend:** ~0–200

By identifying the minimum and maximum values, the system could map the full flex range directly to 0–180° servo movement. This also allowed dynamic constraint of values within a safe operating window, reducing unnecessary overdriving of the motor.



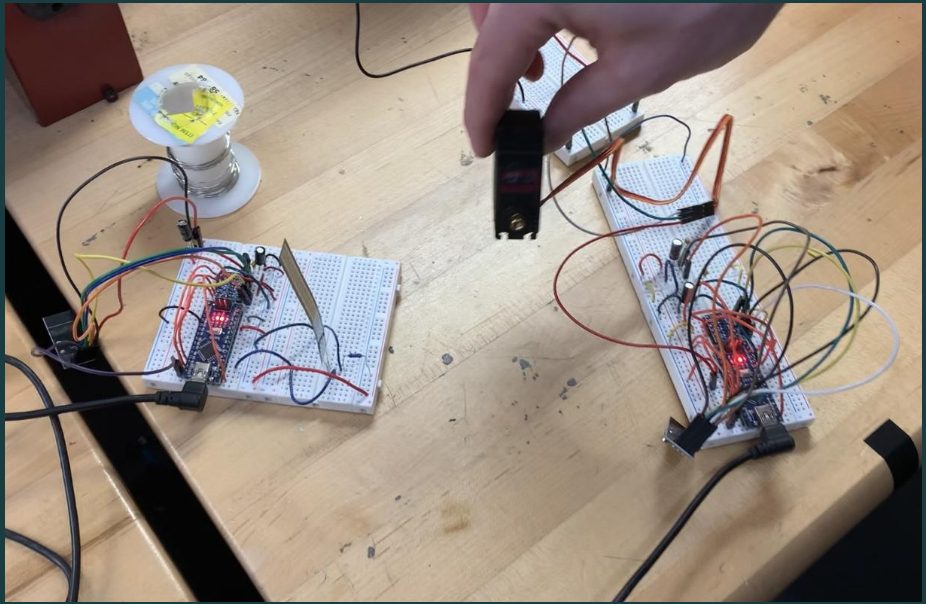
Hand Assembly

- Hand is entirely 3D printed using PLA filament
- Mechanical linkage system for fingers
- Cords will be threaded from servos to base of fingers. Custom bearings let the servo pull in both directions
- PCB to be mounted to back of forearm
- Fingers screwed in place to prevent splaying



Prototyping and Testing

- To verify functionality, serial values from flex sensor is monitored
- Values are used to fine tune algorithm that converts the range of flex sensor to range of servo
- One set of sensor to servo communications has been verified to work, further tests will increment to verify stability



Project Expenses

Part	Cost	Part Status	Budget
ZD10-100 Flex Sensor	$8.81	Chosen/Received	
MCP1700 3.3V regulator (10pcs)	$8.99	Chosen/Received	
MG996r Servo Motors (5pcs)	$20.99	Chosen/Received	
nRF24l01 (4pcs)	$7.89	Chosen/Received	
BSS138 Bidirectional Logic Level Shifter	$7.49	Chosen/Received	
Servo Power Supply		Not Chosen/Pending	
Arduino Nano (2 pack)	$11.99	Chosen/Received	
Miady portable charger (2 pack)	$17.99	Chosen/Received	
PCBs	$129.42	Chosen/Ordered	
Total Current Expenses	$230.82		$500

Team roles and work distribution

	Hand	Glove	Software	PCB Design	CAD/Printing	RF Comms	Power
Rian			✔			✔	
Kane		✔			✔		
Christopher	✔		✔		✔		
Griseld				✔			✔

Timeline

Task	Start Date
PCB Assembly	3/2/26
PCB Testing	3/3/26
Complete shift from breadboard to PCB	3/5/26
Full servo power test	3/10/26
Full RF comms tests	3/14/26
Full assembly and diagnostic tests	3/18/26
Final Demo	4/21/26

- PCB revision may be required but will depend on outcome of testing
- Around a week delay for tasks after power test should PCB need to be revised

Thank you