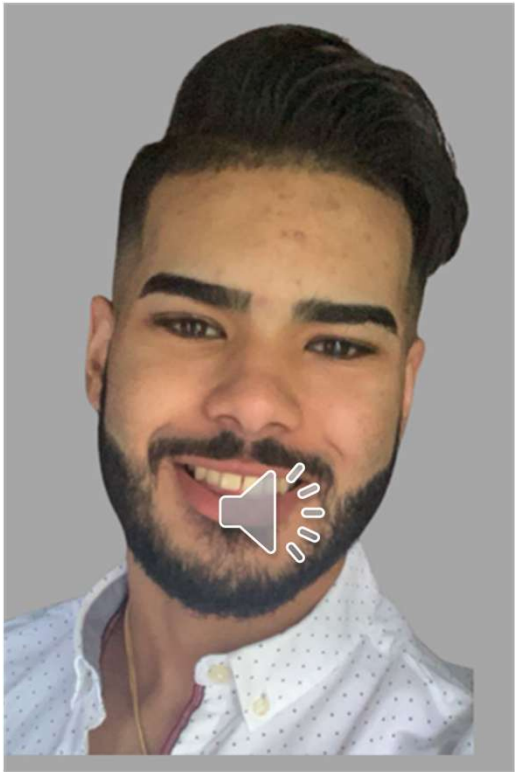


A close-up, slightly blurred photograph of a piano keyboard and the internal hammer mechanism. The keys are white and black, and the hammers are visible in the background. The image has a warm, brownish tint.

ParaPedal: A Practical Product for Paraplegic Piano Players!

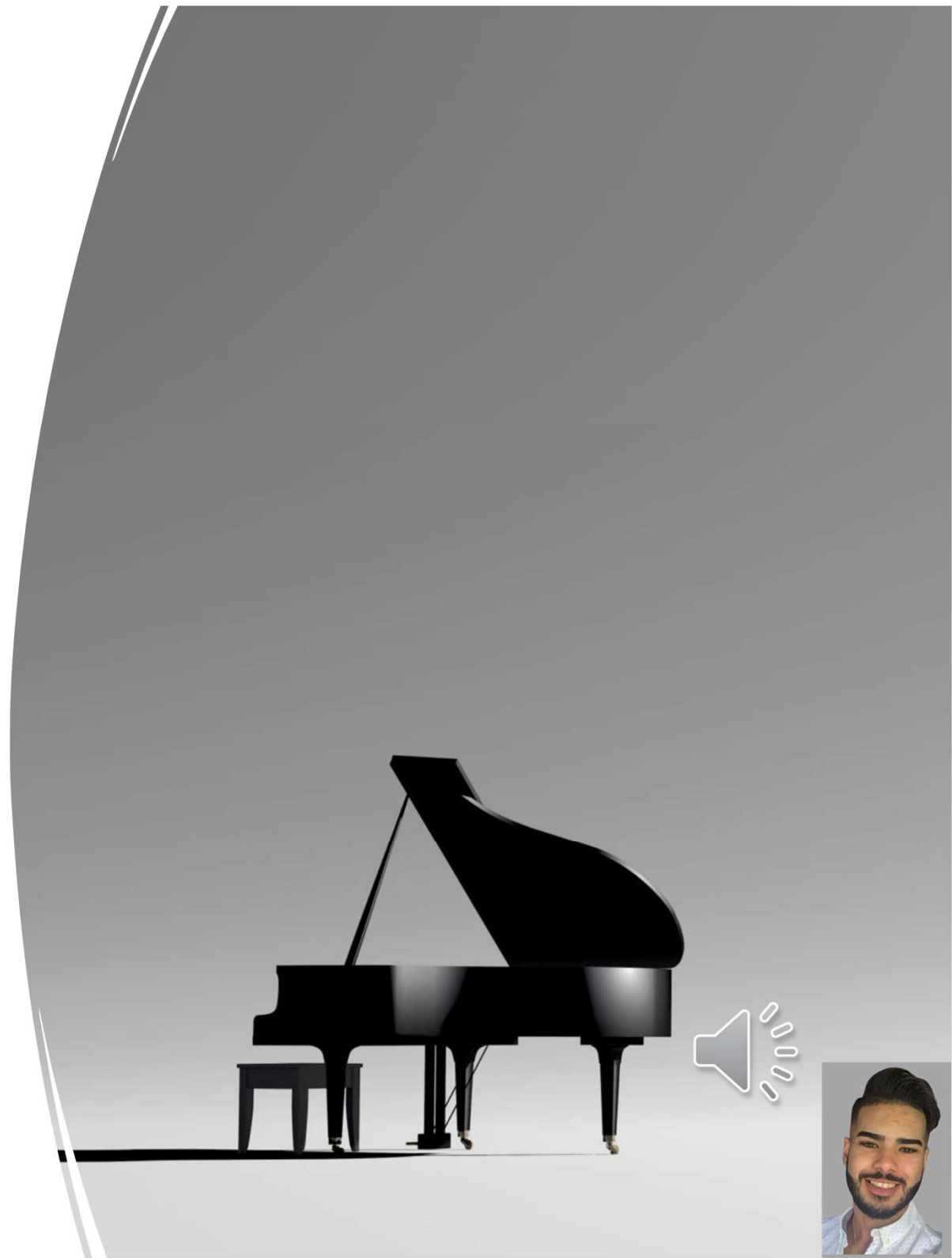
Group 15 – Senior Design
University of Central Florida





Motivation & Background

- Inspired by teammate Benedict, a pianist in a wheelchair
- Problem: Cannot use legs to press piano pedals
- Few assistive devices exist; bulky and limited



Goals



Basic: Sustain pedal,
intuitive use, no piano
modification



Advanced: 3 pedals,
quiet motor actuation



Stretch:
Breath/muscle
input, phone
app calibration
haptic feedback



Objectives

Quantitative Objectives

Motor System

Motor Response Time (from user input)	500-700 Milliseconds
Maximum Power Consumption	≤ 20 Watts

User Input (Button/Dial)

Press Detection Threshold	5-7 Newtons
Pressure Sensor Recognition Time	≤ 500 Miliseconds

Control Unit

Voltage Input	3.3-5 Volts
Maximum Current Draw	< 500 MiliAmps

Overall System

Device Weight	≥ 15 Kilograms
Battery Runtime	≤ 1 Hour
Communication Range	≥ 4 meters
Response Time	≤ 2 Second
Accuracy	75-85%
System Setup Time	≤ 2 Minutes





Engineering Specifications



House of Quality

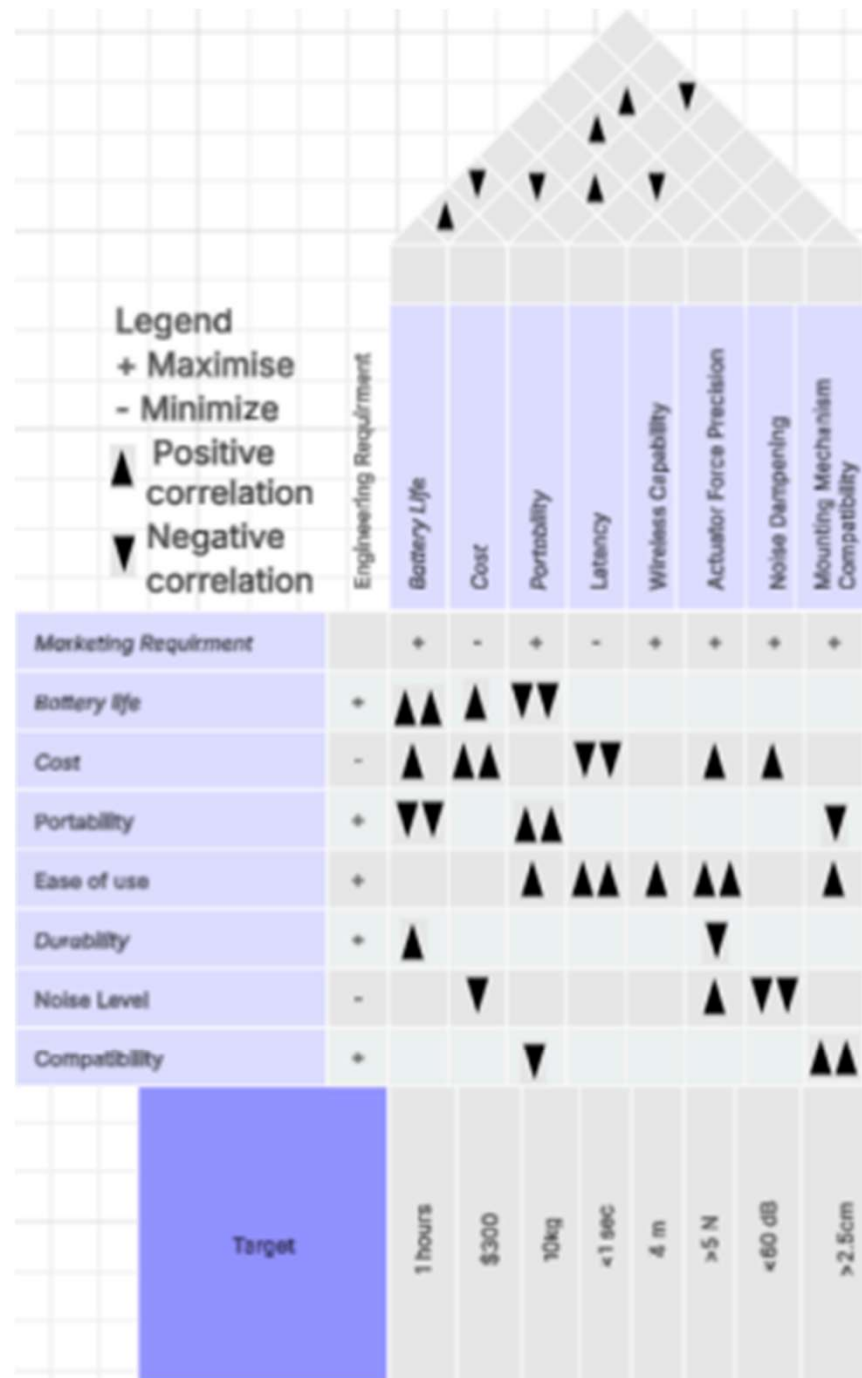
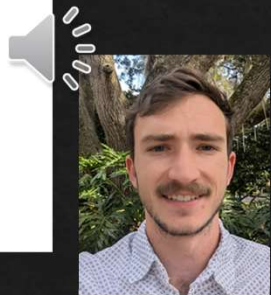
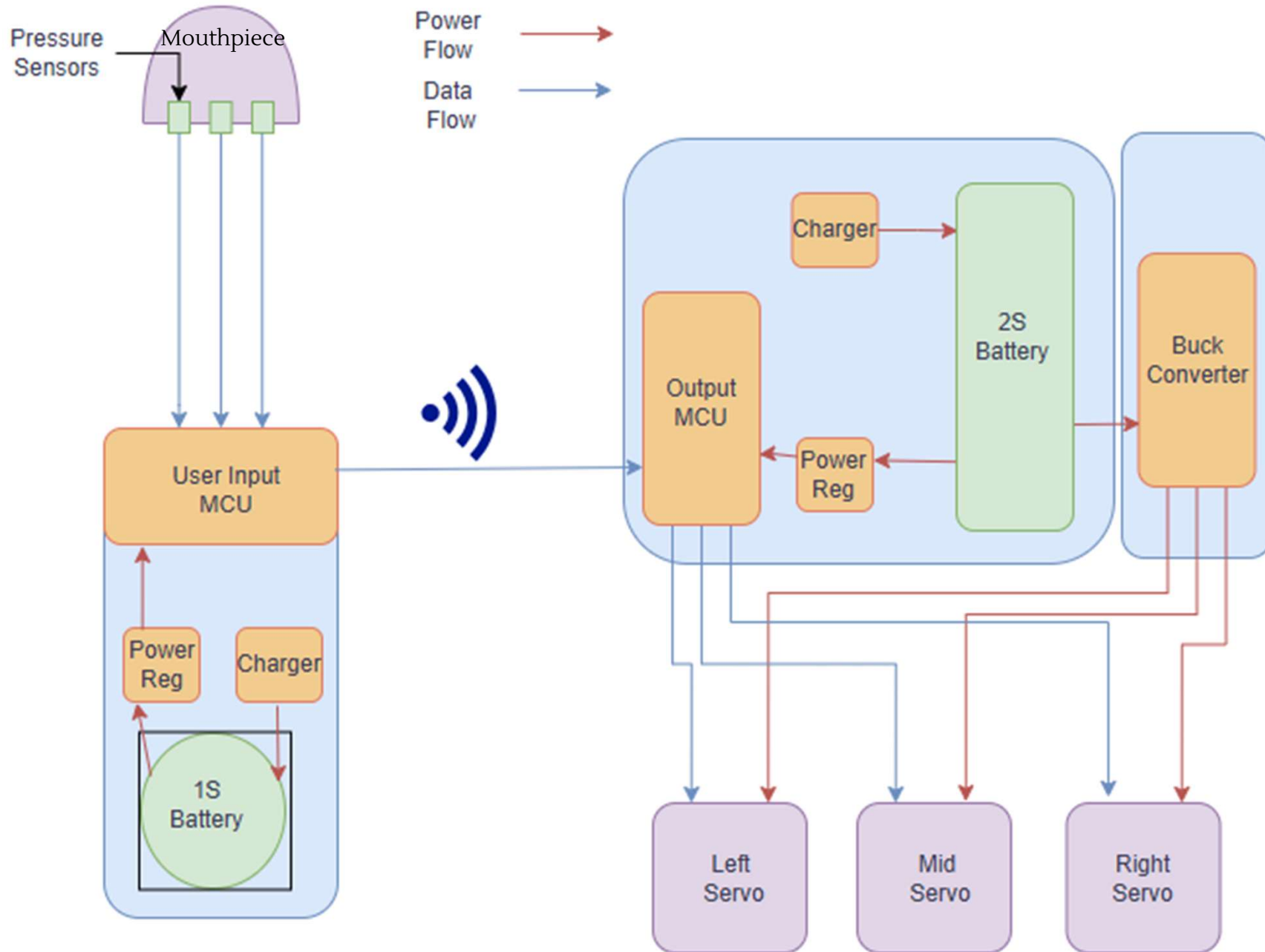




Image of Product



System Diagram



Hardware Selection Process

Categories:
Mouthpiece,
Motor, MCU,
Power



Compared
multiple
technologies for
each



Chose final parts
based on
performance, cost,
usability



Mouthpiece & Sensor



Mouthpiece & Sensor

OPTION	PROS	CONS	FINAL CHOICE
BBTO Sports Guard	Cheap, easy to modify	Bulky, not discreet	✗
Goyo Night Guard	Comfortable, discreet, sleek	More expensive	✓ Final Mouthpiece
Capacitive Sensor	Non-contact sensing	Moisture sensitive, complex integration	✗
RP-S5 Force Sensor	Biocompatible, compact, simple analog signal, 1–10 N range	Limited range	✓ Final Sensor



Actuator / Motor



Actuator vs Motor

Option	Pros	Cons	Final Choice
Linear Actuator (20N type)	Direct linear motion	Bulky, slow, noisy, higher cost	✗
Linear Actuator (VEVOR 1500N)	Very high force capability	Overkill, heavy, slow, expensive	✗
Servo Motor – MG995	PWM control, decent torque	Lower torque/precision than MG996R	✗
Servo Motor – MG996R	13 kg·cm torque, ~0.17s/60°, compact & quiet	Limited rotation	✓



Microcontroller



Final: ESP32 microcontroller



Integrated Wi-Fi + Bluetooth



Multiple ADC inputs and PWM outputs



Low power,
Arduino compatible



Microcontroller

Option	Pros	Cons	Final Choice
Nordic	Good wireless performance	Limited I/O	✗
STM32	Powerful, flexible	Complex, higher learning curve	✗
ESP32	Wi-Fi + Bluetooth, ADCs, PWM, low power	None significant	✓



Power Systems

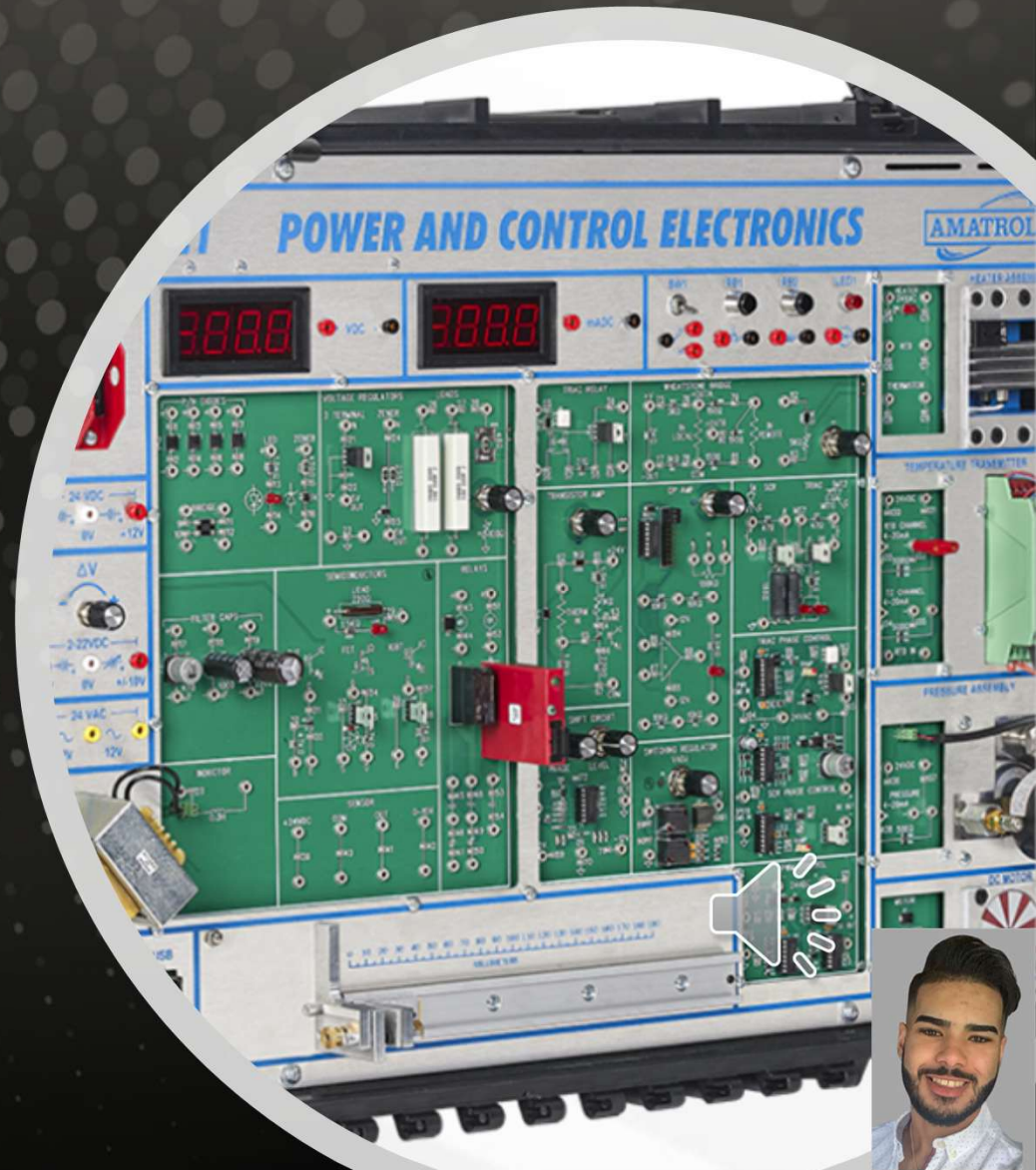
Input Board: LIR2450 coin cell battery

Motor Board: 2×18650 battery pack

Regulation: MP9928GF-Z regulator

MOSFET: SIR426DP

Charger: BQ25887



Power Supply Unit Components

Option	Pros	Cons	Final Choice
LIR2450 Coin Cell	Small, lightweight, rechargeable	Limited capacity	✓ (Mouthpiece Input Board)
2×18650 Pack	High current supply, good capacity	Bulkier, needs balancing	✓ (Motor Board)
MOSFET – SIR426DP	Low Rds(on), efficient switching	Requires driver circuit	✓
Charger – TPB4056A20	Cheap, basic protection	Limited features, 1-cell only	✗
Charger – BQ25887	USB input, 2-cell support, safe charging	Slightly higher cost	✓



Regulators

Option	Pros	Cons	Final Choice
LDO (LP2966)	Simple, easy to use	3.3V only, low current (not for motors)	✗
XC6204B502MR-G	Flexible voltage options	Cannot supply motor currents (<1A)	✗
MP2338	>90% efficiency, up to ~3A	Near limits under motor transients	✗
MP9928GF-Z	>90% efficiency, handles motor spikes, ~3A+	Larger than an LDO	✓



Final Hardware Summary

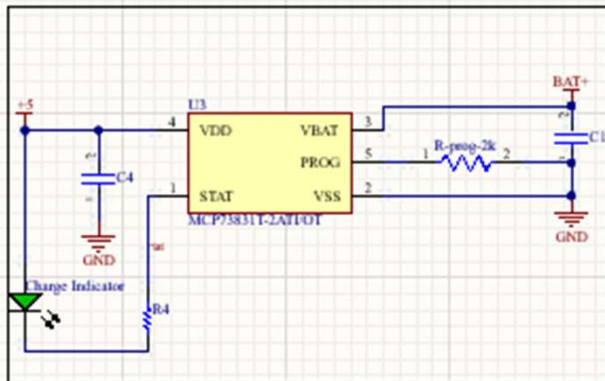
Subsystem	Final Choice	Why
Power	LIR2450 + 2×18650 + MP9928GF-Z + BQ25887+SIR42 6DP	Light, efficient, safe
Mouthpiece	Goyo Night Guard	Comfortable, sleek, hidden
Sensor	RP-S5 Force Sensor	Simple, reliable, safe
Display	SPI TFT LCD (1.8")	Clear, cheap, Arduino compatible
Actuator	MG996R Servo	High torque, fast, affordable
MCU	ESP32	Wi-Fi + Bluetooth, flexible, low power



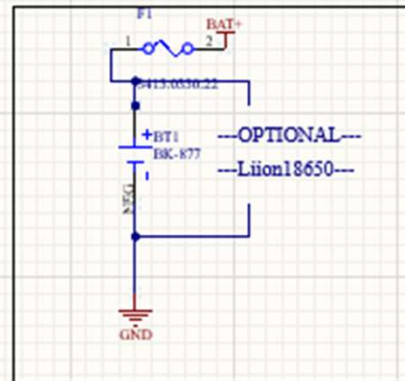
User Input Power Regulation

Power Path

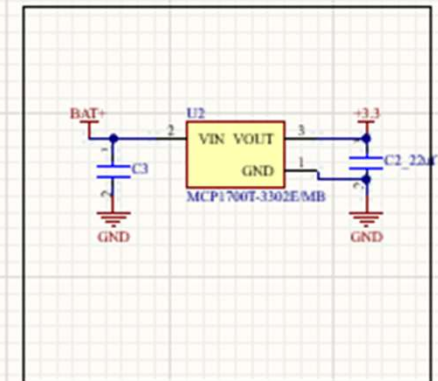
Battery Charger IC - 5VUSB input



1S Lithium Battery LIR2450 or 18650

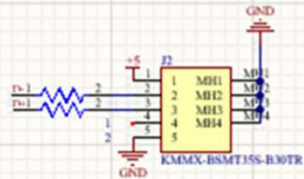


+3.3V Regulator

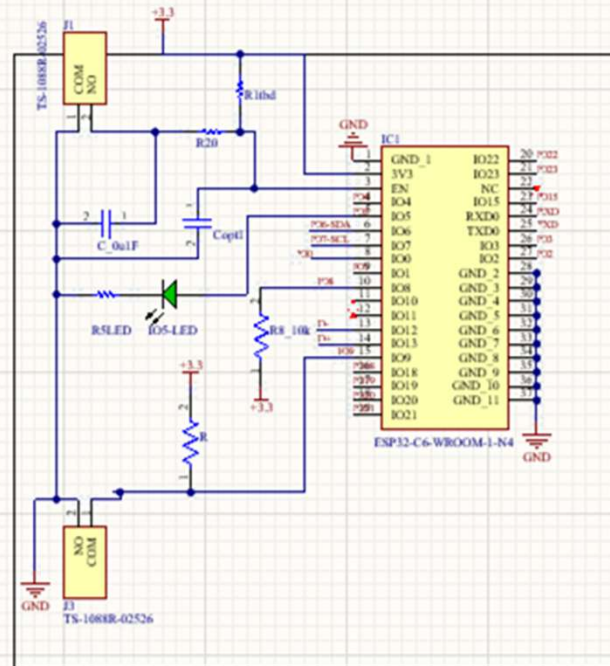


User Input Control

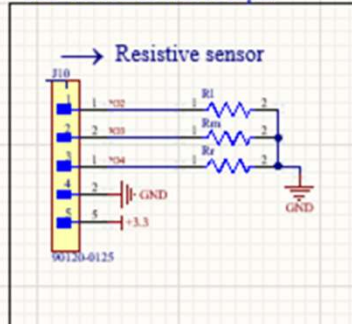
MICROUSB input (programming and charging)



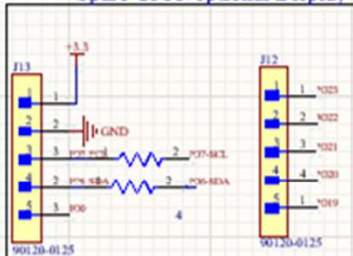
MCU (ESP32-WROOM-C6)



3ch -Pressures sensor Input



Spare GPIO/optional Display



Info	Pass
Start	A1

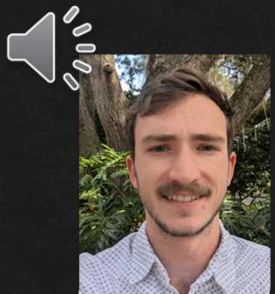
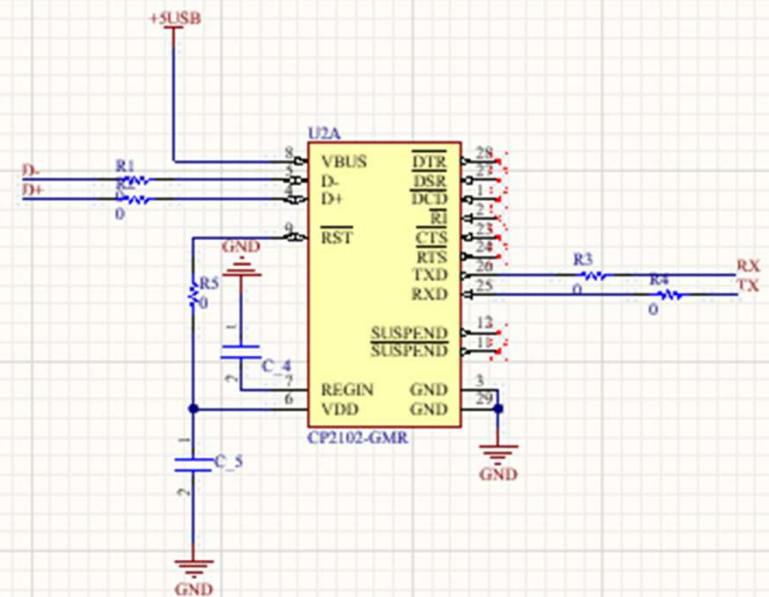
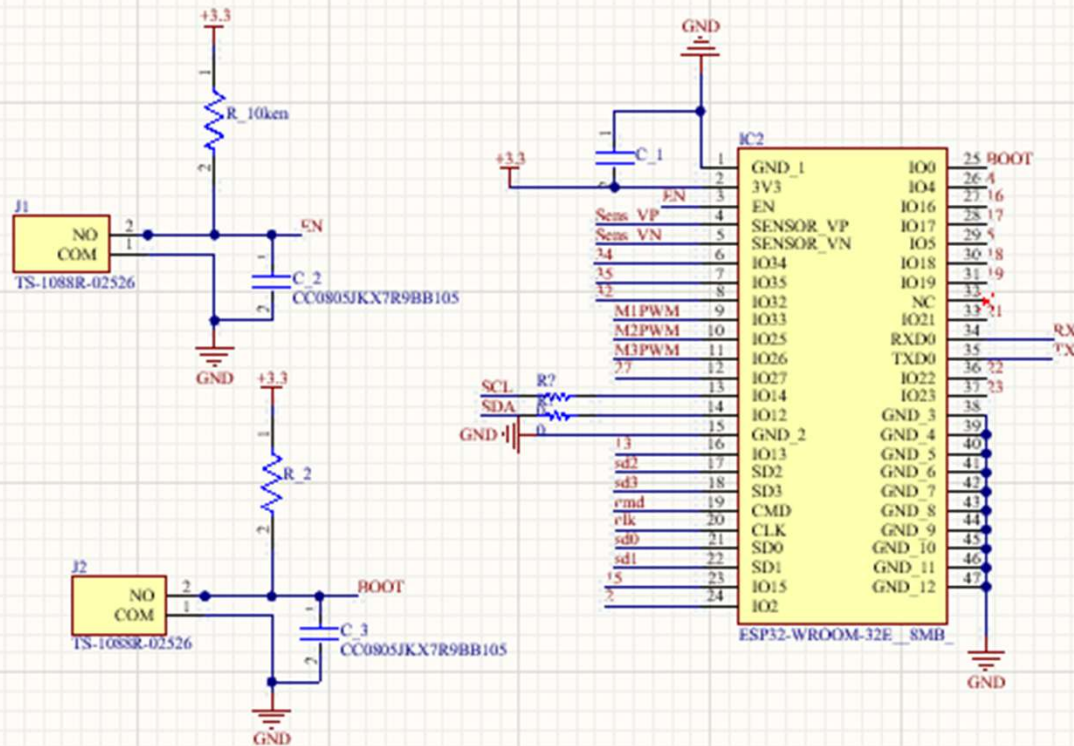


The image displays a complex PCB layout for a sensor module. The design is organized into several functional blocks:

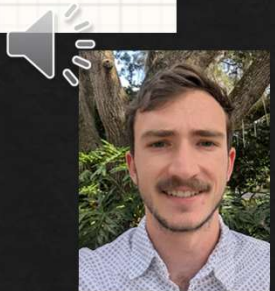
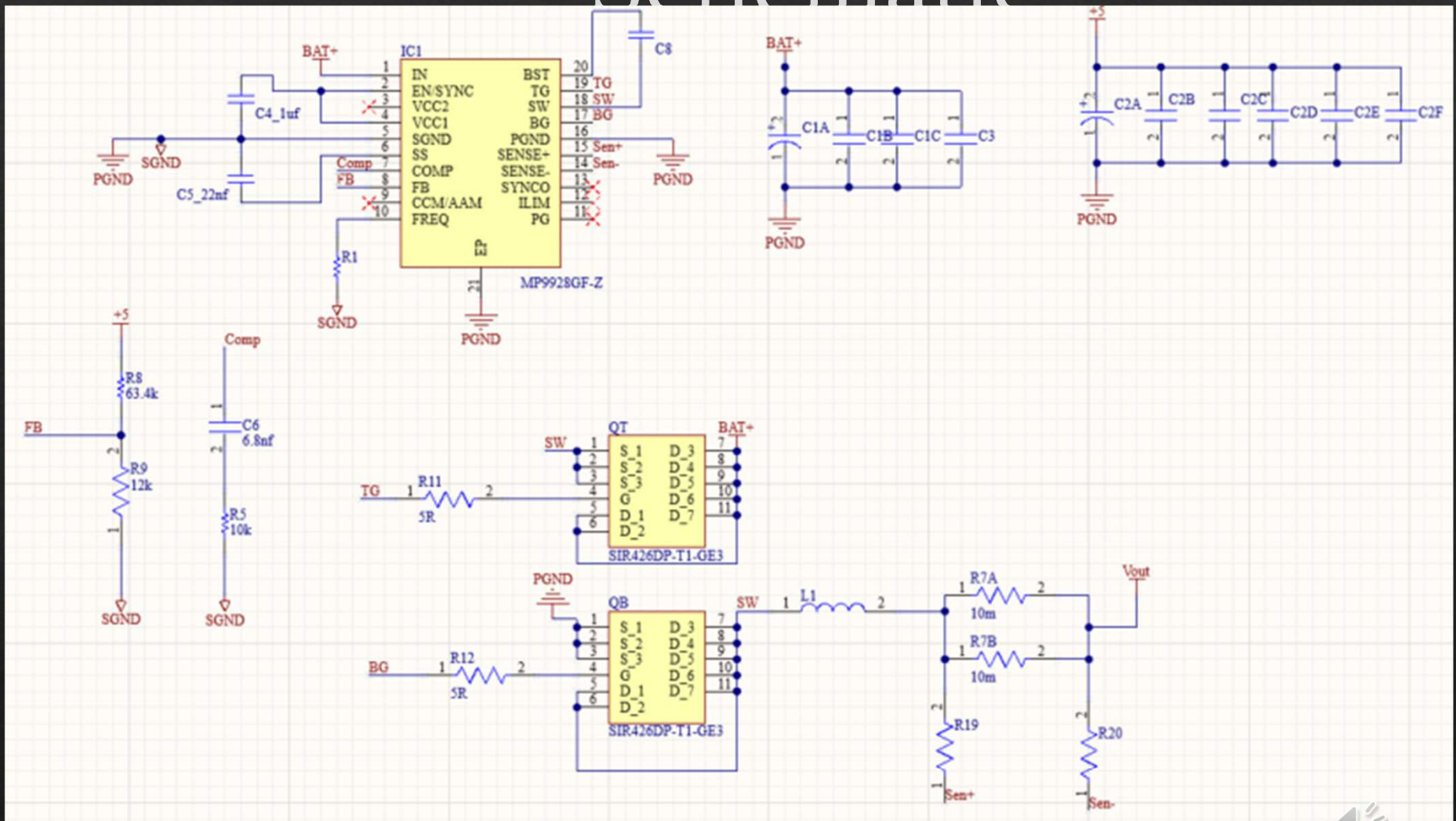
- USB Interface:** A USB connector (J3) is located at the top left, with pins for SHLD6, SHLD5, SHLD4, SHLD3, SHLD2, SHLD1, GND, ID, D+, D-, and VBUS. A +5V USB supply is connected to the VBUS pin.
- Power Management:** A +3.3V regulator (AP63203WU-7) is used to step down the +5V USB supply to +3.3V. It includes input capacitors (C20, C40), an output capacitor (C30), and a feedback network (R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13, R14, R15, R16, R17, R18, R19, R20, R21, R22, R23, R24, R25, R26, R27, R28, R29, R30, R31, R32, R33, R34, R35, R36, R37, R38, R39, R40, R41, R42, R43, R44, R45, R46, R47, R48, R49, R50, R51, R52, R53, R54, R55, R56, R57, R58, R59, R60, R61, R62, R63, R64, R65, R66, R67, R68, R69, R70, R71, R72, R73, R74, R75, R76, R77, R78, R79, R80, R81, R82, R83, R84, R85, R86, R87, R88, R89, R90, R91, R92, R93, R94, R95, R96, R97, R98, R99, R100, R101, R102, R103, R104, R105, R106, R107, R108, R109, R110, R111, R112, R113, R114, R115, R116, R117, R118, R119, R120, R121, R122, R123, R124, R125, R126, R127, R128, R129, R130, R131, R132, R133, R134, R135, R136, R137, R138, R139, R140, R141, R142, R143, R144, R145, R146, R147, R148, R149, R150, R151, R152, R153, R154, R155, R156, R157, R158, R159, R160, R161, R162, R163, R164, R165, R166, R167, R168, R169, R170, R171, R172, R173, R174, R175, R176, R177, R178, R179, R180, R181, R182, R183, R184, R185, R186, R187, R188, R189, R190, R191, R192, R193, R194, R195, R196, R197, R198, R199, R200, R201, R202, R203, R204, R205, R206, R207, R208, R209, R210, R211, R212, R213, R214, R215, R216, R217, R218, R219, R220, R221, R222, R223, R224, R225, R226, R227, R228, R229, R230, R231, R232, R233, R234, R235, R236, R237, R238, R239, R240, R241, R242, R243, R244, R245, R246, R247, R248, R249, R250, R251, R252, R253, R254, R255, R256, R257, R258, R259, R260, R261, R262, R263, R264, R265, R266, R267, R268, R269, R270, R271, R272, R273, R274, R275, R276, R277, R278, R279, R280, R281, R282, R283, R284, R285, R286, R287, R288, R289, R290, R291, R292, R293, R294, R295, R296, R297, R298, R299, R300, R301, R302, R303, R304, R305, R306, R307, R308, R309, R310, R311, R312, R313, R314, R315, R316, R317, R318, R319, R320, R321, R322, R323, R324, R325, R326, R327, R328, R329, R330, R331, R332, R333, R334, R335, R336, R337, R338, R339, R340, R341, R342, R343, R344, R345, R346, R347, R348, R349, R350, R351, R352, R353, R354, R355, R356, R357, R358, R359, R360, R361, R362, R363, R364, R365, R366, R367, R368, R369, R370, R371, R372, R373, R374, R375, R376, R377, R378, R379, R380, R381, R382, R383, R384, R385, R386, R387, R388, R389, R390, R391, R392, R393, R394, R395, R396, R397, R398, R399, R400, R401, R402, R403, R404, R405, R406, R407, R408, R409, R410, R411, R412, R413, R414, R415, R416, R417, R418, R419, R420, R421, R422, R423, R424, R425, R426, R427, R428, R429, R430, R431, R432, R433, R434, R435, R436, R437, R438, R439, R440, R441, R442, R443, R444, R445, R446, R447, R448, R449, R450, R451, R452, R453, R454, R455, R456, R457, R458, R459, R460, R461, R462, R463, R464, R465, R466, R467, R468, R469, R470, R471, R472, R473, R474, R475, R476, R477, R478, R479, R480, R481, R482, R483, R484, R485, R486, R487, R488, R489, R490, R491, R492, R493, R494, R495, R496, R497, R498, R499, R500, R501, R502, R503, R504, R505, R506, R507, R508, R509, R510, R511, R512, R513, R514, R515, R516, R517, R518, R519, R520, R521, R522, R523, R524, R525, R526, R527, R528, R529, R530, R531, R532, R533, R534, R535, R536, R537, R538, R539, R540, R541, R542, R543, R544, R545, R546, R547, R548, R549, R550, R551, R552, R553, R554, R555, R556, R557, R558, R559, R560, R561, R562, R563, R564, R565, R566, R567, R568, R569, R570, R571, R572, R573, R574, R575, R576, R577, R578, R579, R580, R581, R582, R583, R584, R585, R586, R587, R588, R589, R590, R591, R592, R593, R594, R595, R596, R597, R598, R599, R600, R601, R602, R603, R604, R605, R606, R607, R608, R609, R610, R611, R612, R613, R614, R615, R616, R617, R618, R619, R620, R621, R622, R623, R624, R625, R626, R627, R628, R629, R630, R631, R632, R633, R634, R635, R636, R637, R638, R639, R640, R641, R642, R643, R644, R645, R646, R647, R648, R649, R650, R651, R652, R653, R654, R655, R656, R657, R658, R659, R660, R661, R662, R663, R664, R665, R666, R667, R668, R669, R670, R671, R672, R673, R674, R675, R676, R677, R678, R679, R680, R681, R682, R683, R684, R685, R686, R687, R688, R689, R690, R691, R692, R693, R694, R695, R696, R697, R698, R699, R700, R701, R702, R703, R704, R705, R706, R707, R708, R709, R710, R711, R712, R713, R714, R715, R716, R717, R718, R719, R720, R721, R722, R723, R724, R725, R726, R727, R728, R729, R730, R731, R732, R733, R734, R735, R736, R737, R738, R739, R740, R741, R742, R743, R744, R745, R746, R747, R748, R749, R750, R751, R752, R753, R754, R755, R756, R757, R758, R759, R760, R761, R762, R763, R764, R765, R766, R767, R768, R769, R770, R771, R772, R773, R774, R775, R776, R777, R778, R779, R780, R781, R782, R783, R784, R785, R786, R787, R788, R789, R790, R791, R792, R793, R794,



Output board Control

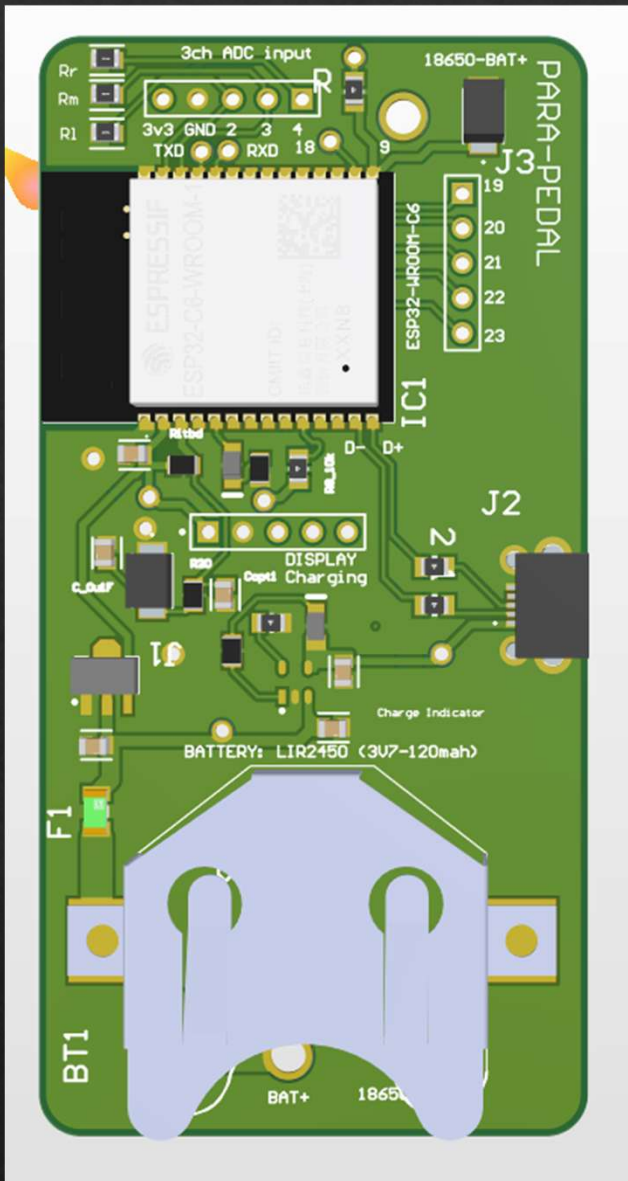


Buck Converter Schematic



PCB implementation





User Input Board

Primary Function

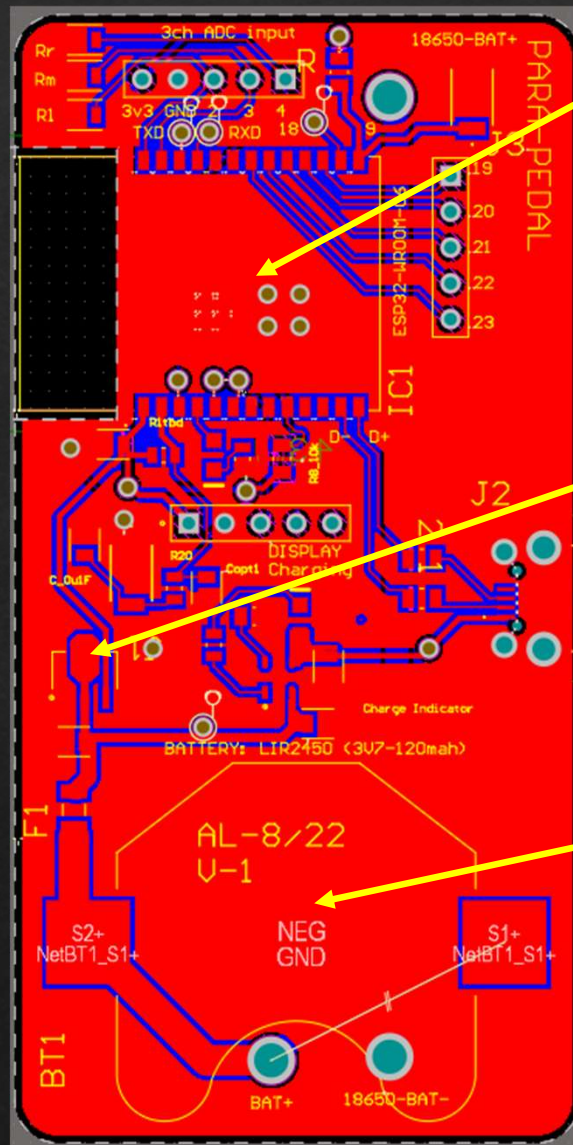
- ◆ To sense the user input
- ◆ Transmit this data to the output controller

Main Components

- ◆ MCU :ESP32 - C6
- ◆ Battery :LIR2450 120mah
- ◆ Charger: MCP73831T
- ◆ Regulator: MCP1700T



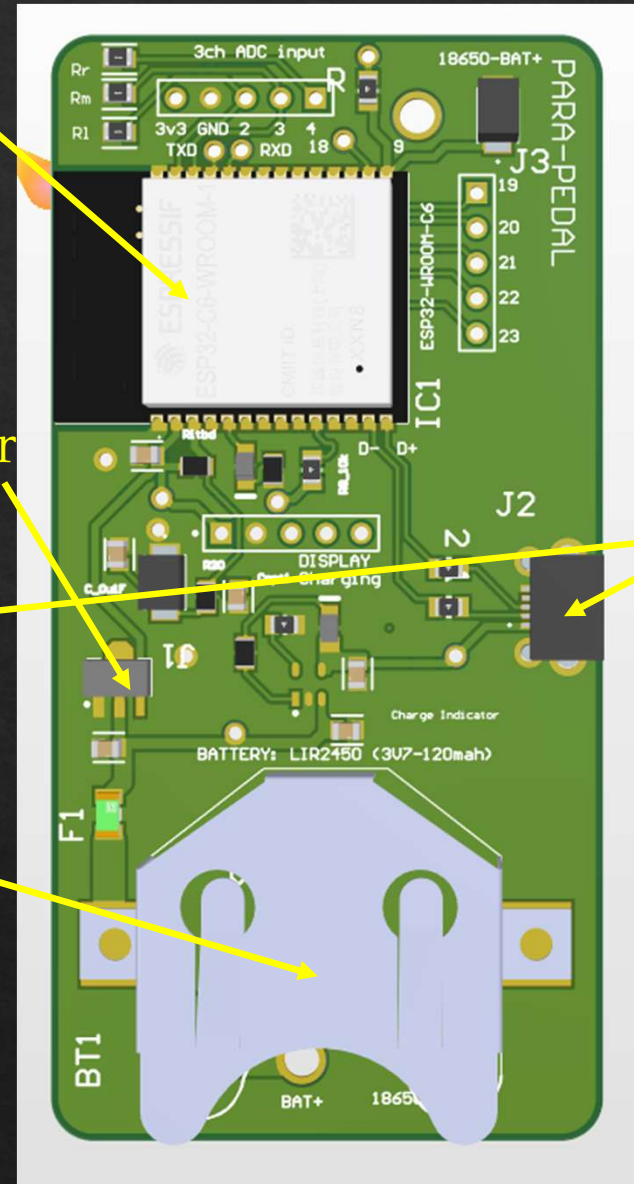
User Input Layout



MCU

Regulator

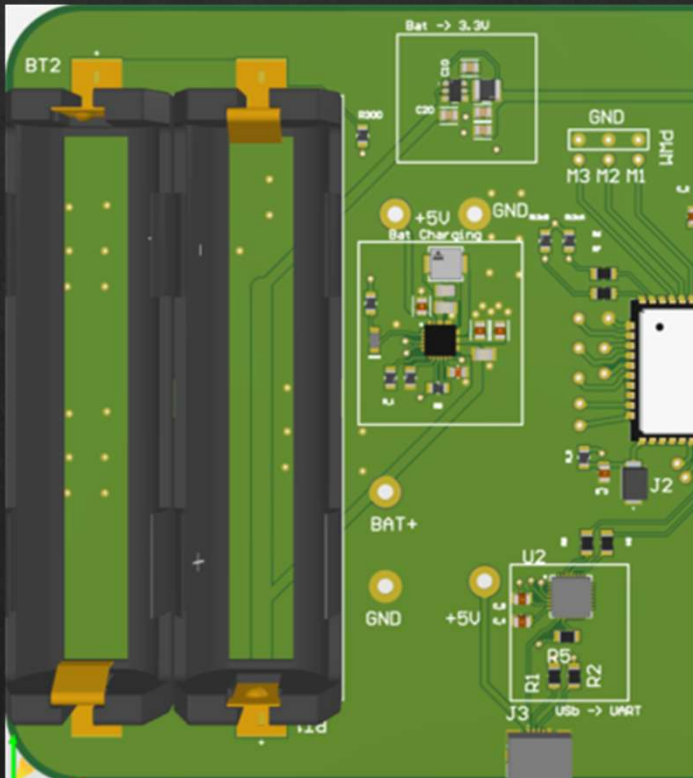
Battery



USB



Output Board



Primary function

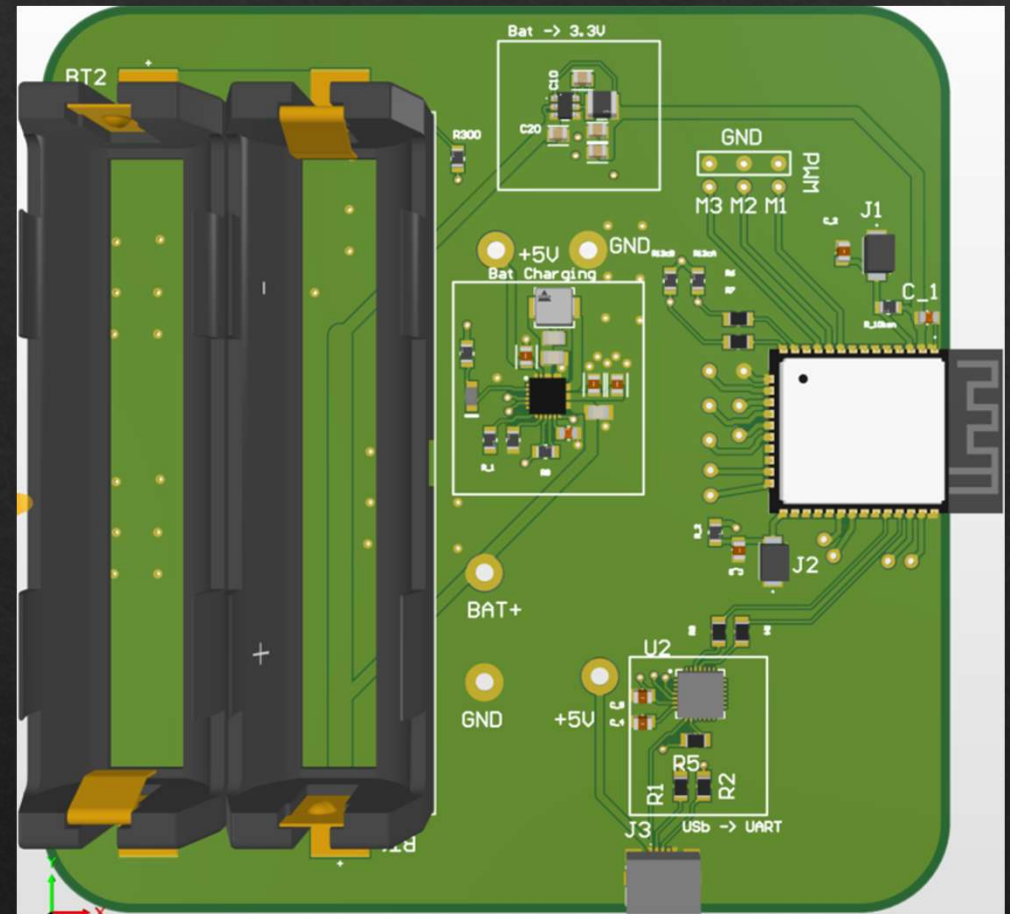
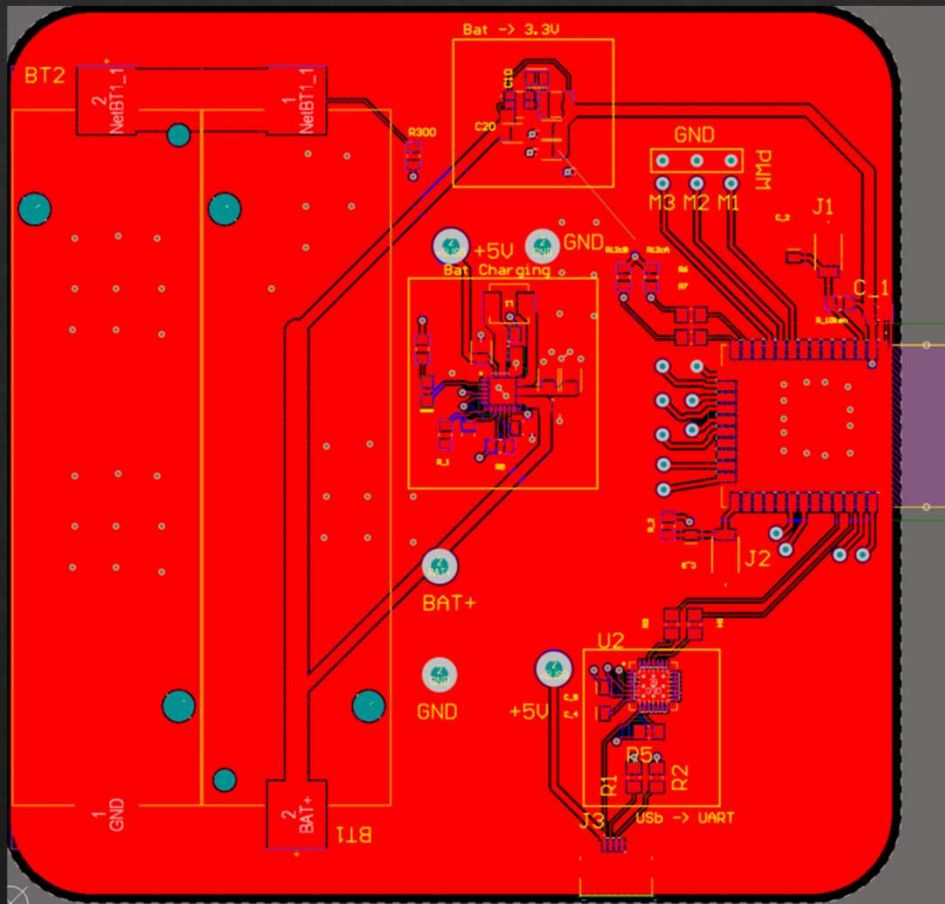
- ◆ to receive data wirelessly from input board
- ◆ Output PWM to the servo for actuation

Main Components

- ◆ MCU :ESP32 -32E
- ◆ Battery :2 X 18650 2200mah
- ◆ Charger: BQ25887RGER
- ◆ Regulator: AP63203WU-7



Output Layout



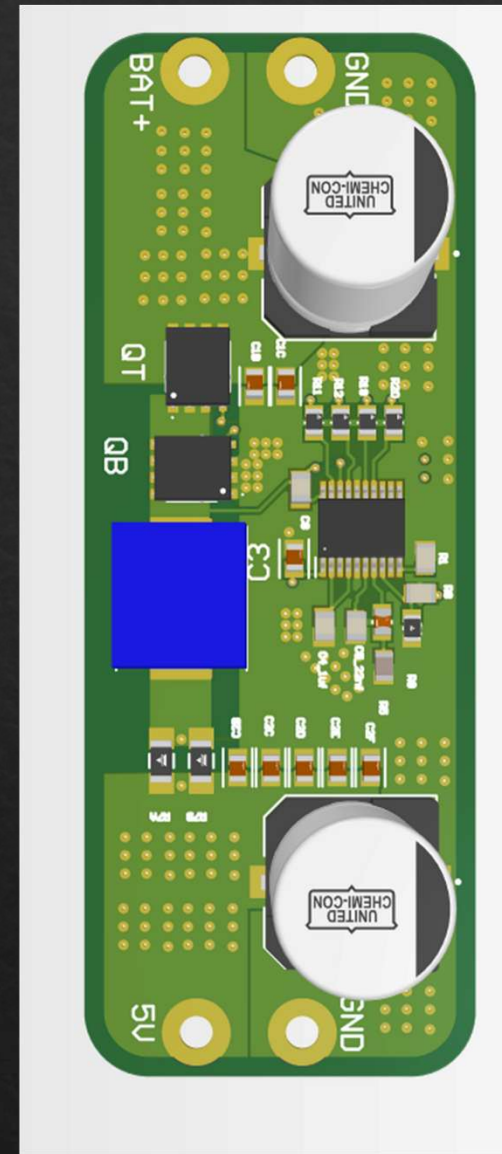
Servo Buck Converter

Primary function

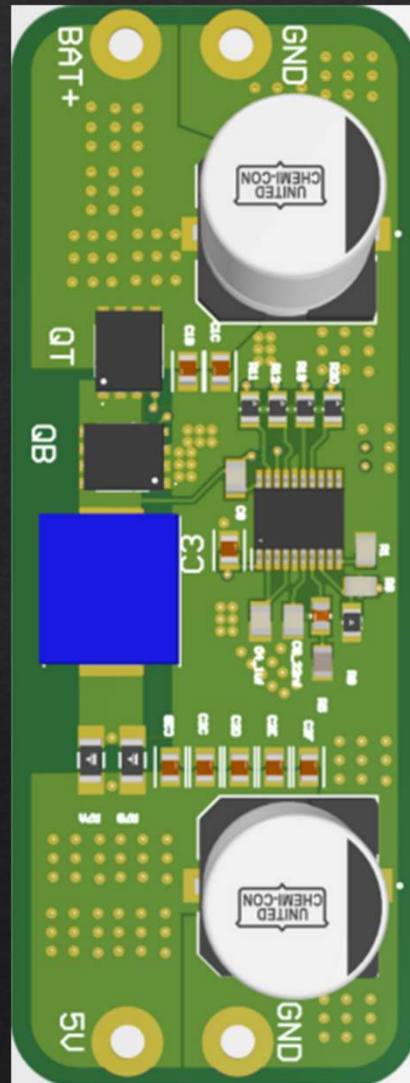
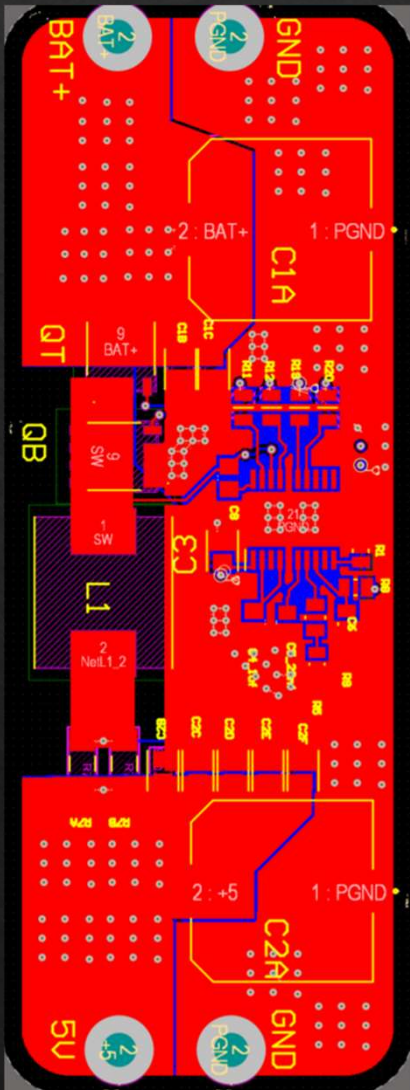
- ◆ Provide sufficient current capacity to the Motors at 5V from Battery source
- ◆ Total current = $2.5A \times 3 = 7.5A$

Main Components

- ◆ Buck controller : MP9928GF-Z
- ◆ Hi/Lo Mosfets: SIR426DP-T1-GE3
- ◆ Inductor : SRP1050WA-3R3M



Buck converter Layout



Wireless Protocols



Wifi	Bluetooth	ESP-now
Great complex protocol handling	Ok complex protocol handling	Poor complex protocol handling
Longer range 50-100m	Short range 10-40m	Good range 50-100
Ok Ability to handle multiple devices	Great Ability to handle multiple devices	Low Ability to handle multiple devices
Good for high throughput	Great for short bursts of information	Only small data packets
Higher power consumption	Low power consumption	Lower power consumption
High Latency	Ok Latency	Low Latency

Wired Protocols

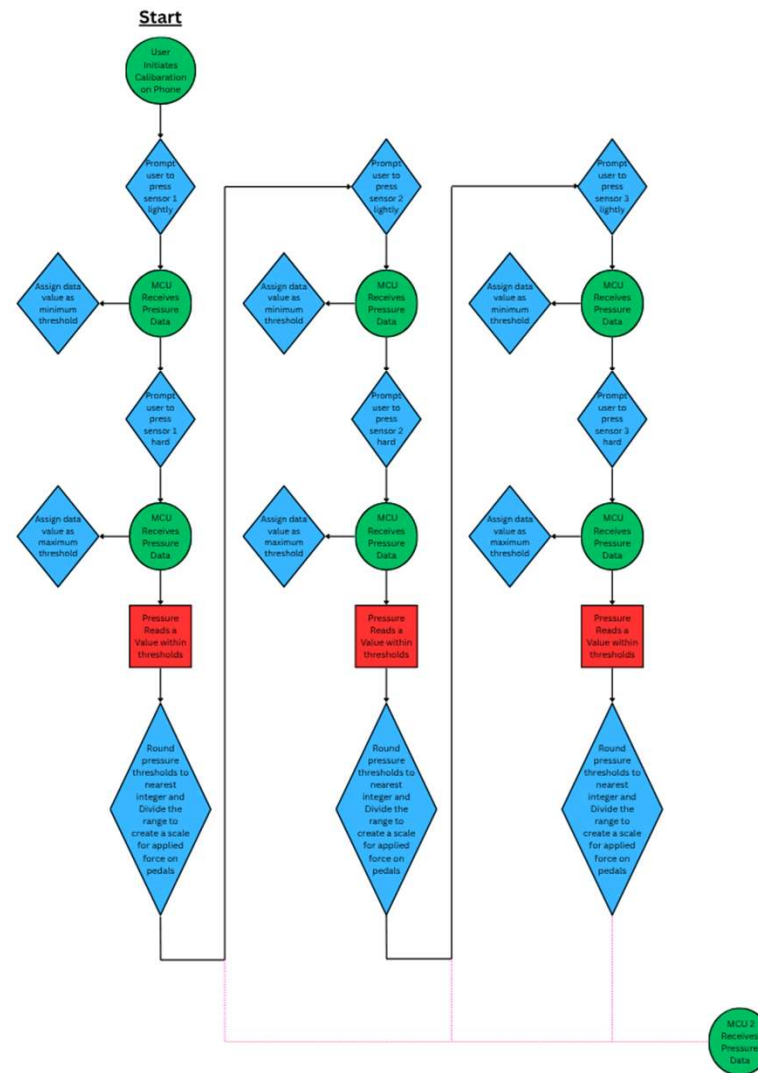


Wifi	Bluetooth	ESP-now
Great complex protocol handling	Ok complex protocol handling	Poor complex protocol handling
Longer range 50-100m	Short range 10-40m	Good range 50-100
Ok Ability to handle multiple devices	Great Ability to handle multiple devices	Low Ability to handle multiple devices
Good for high throughput	Great for short bursts of information	Only small data packets
Higher power consumption	Low power consumption	Lower power consumption
High Latency	Ok Latency	Low Latency

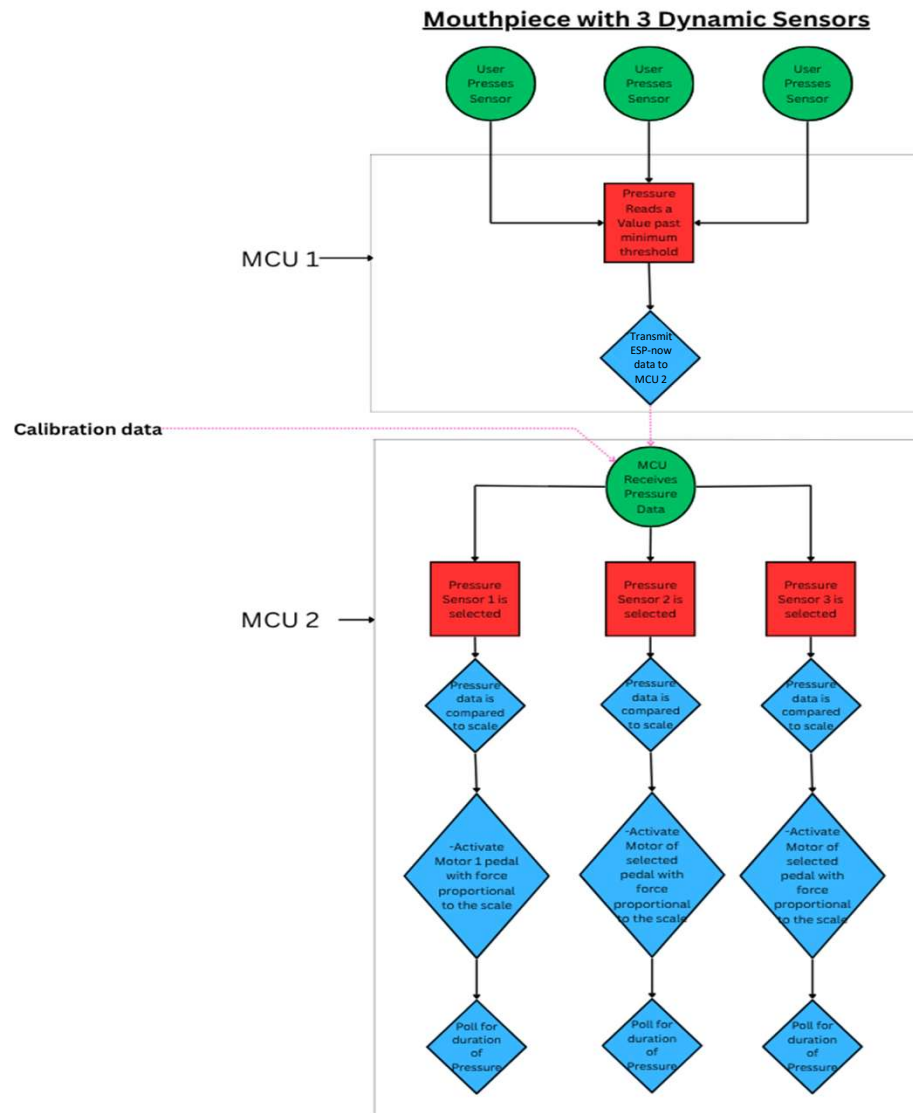
Software design



Calibration



Software design



Selection of Software



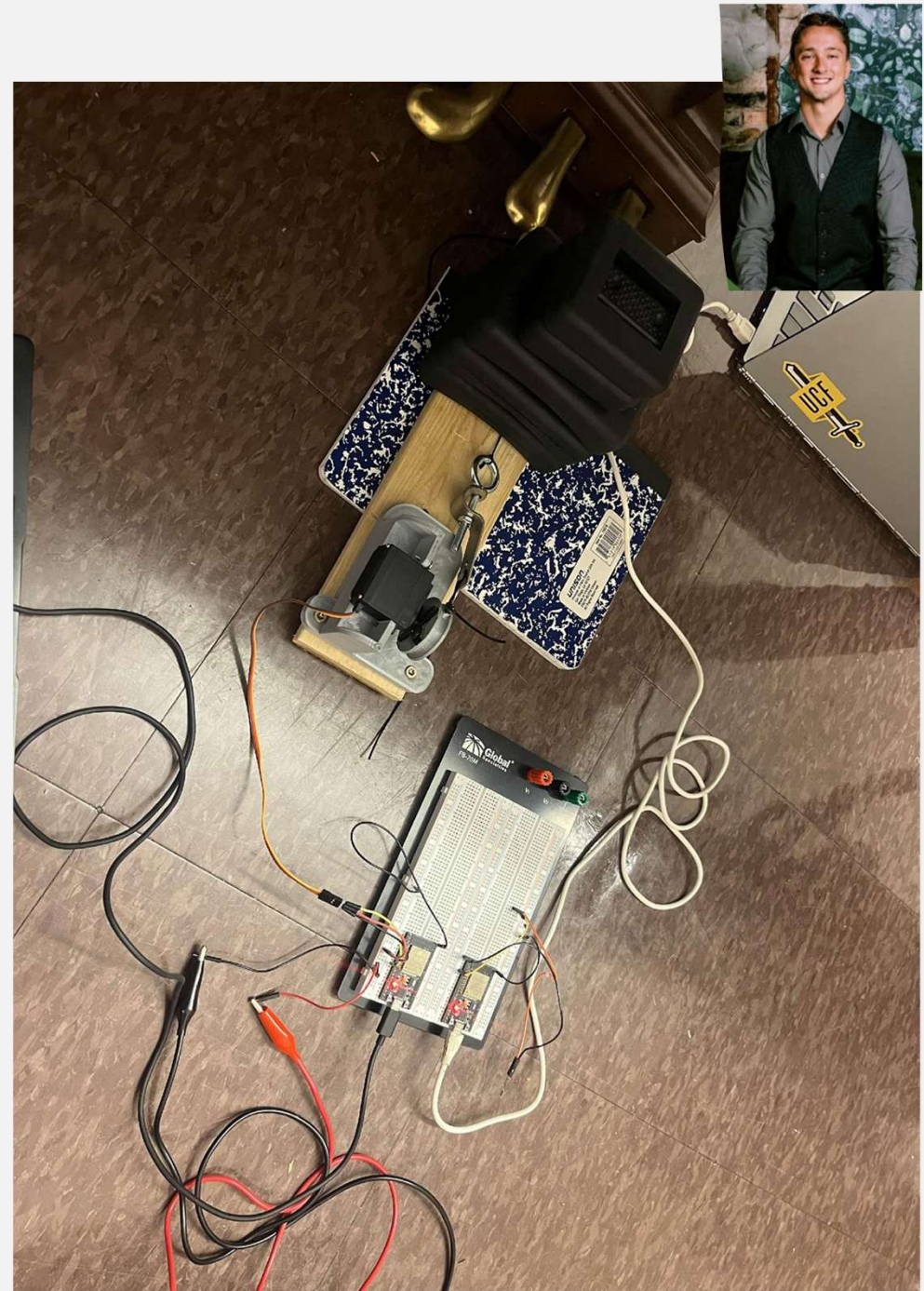
FreeRTOS	Arduino
Good multitasking	Limited multitasking (single-threaded)
Deterministic RTS	Non-deterministic RTS
Good Hardware support	Good Hardware support
High RAM usage	Low overhead
Smaller community libraries	Strong Community libraries

Selection of Software



Python	C++
Vast and easy to use ecosystem	Complex Syntax
Better memory management	User-managed memory
Slower communication	Faster communication
Abstract hardware access	Direct Hardware control
Higher memory usage	Lower memory usage

Prototype Testing



Evaluation Methodology

- **Functional Testing:** Verifying if the system fulfills specifications and requirements
- **Quantitative Benchmarks:** Measuring timing, force, latency, battery life, and wireless range.
- **User Experience Evaluation:** Feedback from our teammate Benedict who has paraplegia in terms of comfort, usability, and ease of learning.



SYSTEM CALIBRATION

Actuator Travel Limits

SET MINIMUM POSITION

SET MAXIMUM POSITION

Pressure Sensor Range

CALIBRATE MINIMUM INPUT

CALIBRATE MAXIMUM INPUT

RETURN TO AUTO MODE

FACTORY RESET

CALIBRATION PARAMETERS

Range: 0Å° - 180Å° (500-2400µs)
Range: 0.0% - 100.0%

System Status

RAW SENSOR DATA

55

AGE (MS)

2422

DATA LATENCY (MS)

A0:B7:65:6

SENSOR AD



Control Software Testing

1435

ACTUATOR POSITION (Å°)

AUTO

OPERATION MODE

LIVE TELEMETRY

Pressure Sensor
MAC Address: A0:B7:65:6
> Adding peer with MAC: A0:B7:65:6
> Peer added successfully
> Setup complete. Starting press
> Pressure: 43.0%, ADC: 1787, Sen
> Pressure: 44.0%, ADC: 1813, Sen
> Pressure: 44.0%, ADC: 1831, Sen
> Pressure: 45.0%, ADC: 1856, Sen
> Pressure: 47.0%, ADC: 1953, Sen
> Pressure: 47.0%, ADC: 1947, Sen
> Pressure: 47.0%, ADC: 1927, Sen
> Pressure: 46.0%, ADC: 1919, s
> Pressure: 44.0%, ADC: 1834
> Pressure: 44.0%, ADC: 1834
> Pressure: 46.0%, ADC: 1834

The PWM generation system was tested by simulating sensor input values via software variables.

Motor responded immediately, transitioning smoothly between "pedal on" and "pedal off" states.

Logic loop and signal pathways from controller to servo proven functional.

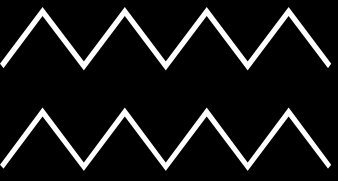
MANUAL OVERRIDE CONTROL

49:45.424 > Loaded calibration
49:45.518 > AP IP address: 192.168.4.1
49:45.518 > MAC Address: A0:B7:65:6
49:45.524 > Use this MAC address
49:45.528 > Web server started
49:45.530 > Open http://192.168.4.1
49:45.534 > Setup complete. Waiting
49:45.557 > Received - Pressure: 0.0%
49:45.699 > Received - Pressure: 0.0%
49:45.854 > Received - Pressure: 0.0%
49:46.006 > Received - Pressure: 0.0%
49:46.148 > Received - Pressure: 0.0%

REAL-TIME MODE ACTIVE

Position updates continuously while adjusting

TARGET: 90Å°



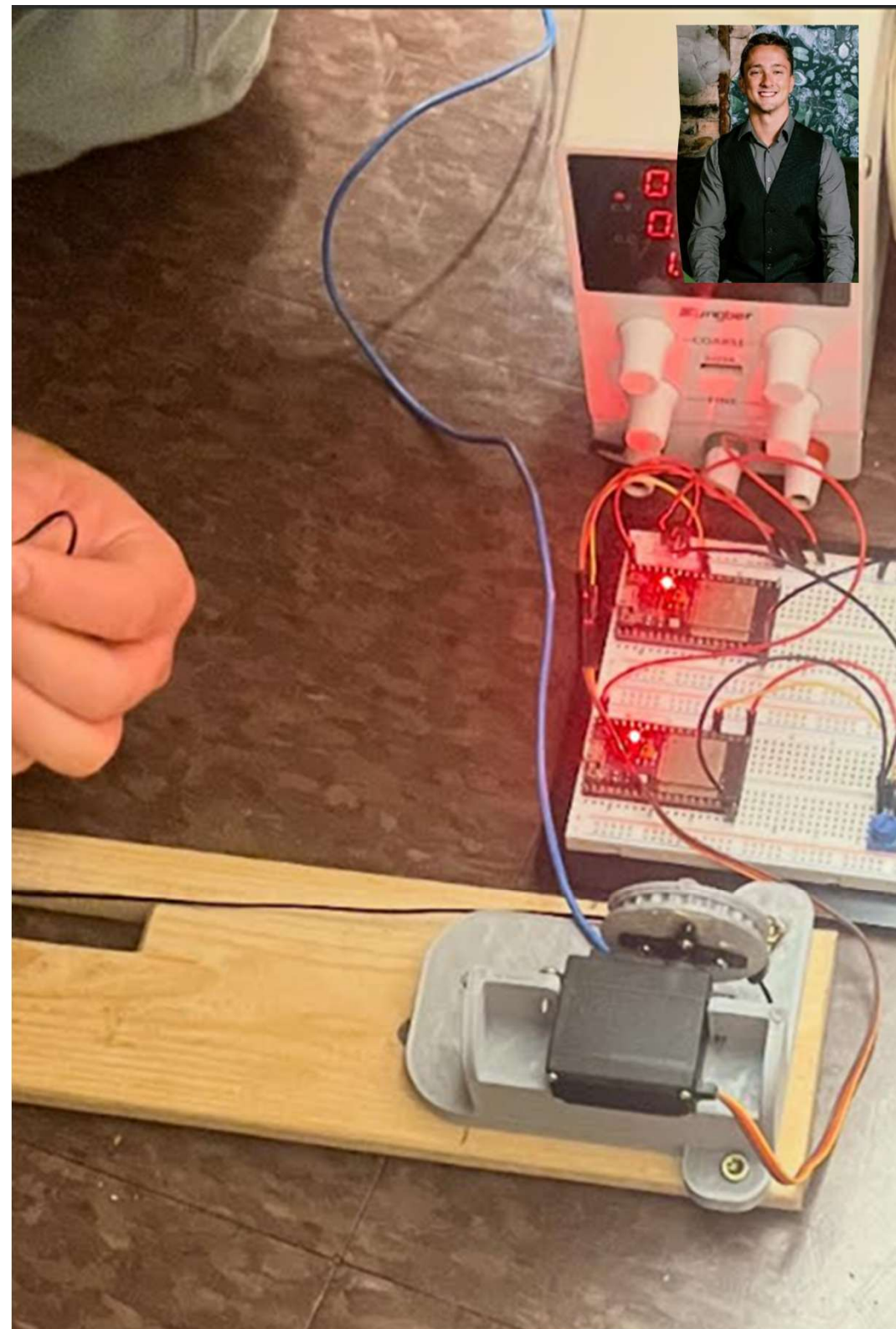
Software Bugs & Solutions



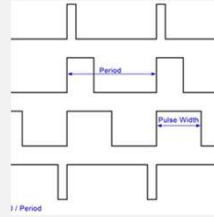
- **Problem 1:** Wasn't able to transmit the full range of the users' input.
- **Solution:** Replaced AD_6db with ADC_11db, increasing our sensor voltage from 1.1V to 3.3V.
- **Problem 2:** Servo Motor twitching, nearly unresponsive.
- **Solution:** Decreased the PWM frequency from 80Hz to 50Hz.

Motor and Structural Testing (1st trial)

- 3D-printed the servo mounting and string pulley system.
- ESP32 WROOM Development Boards connected to laptops.
- Adjustable DC Power Supply for motors.
- Potentiometer used in lieu of pressure sensors



Trial 1 Successes



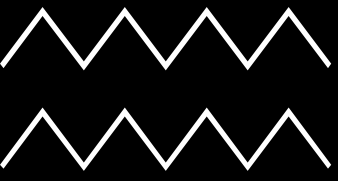
Motor Control Logic:
consistently produced smooth,
responsive, pedal movement.



Mechanical Response: Moved
the pedal with an acceptable
delay



Wireless
Communication:
Signals were properly
sent and received
between boards



Trial 1 Challenges



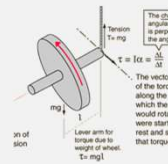
- Upward Lift: When the motor pulled the pedal via the pulley, the base of the device lifted off the floor. This was due to excessive pulling force not being counteracted by sufficient downward weight or anchoring.
- Wheel Displacement: The guiding wheel occasionally rotated out of alignment. This was primarily due to unbalanced torque during actuation.
- Poor Centering: The wheel wasn't perfectly centered with the pedal during setup. This misalignment caused uneven pulling and added sideways drag, making actuation less consistent.

Motor and Structural Testing (2nd trial)

- Added a stronger, upgraded wheel for better torque
- Screws added to secure connection between the wheel and motor shaft
- Custom guiding wheel added for the pulley system stability.
- Weights added to base to mitigate the lifting effect.



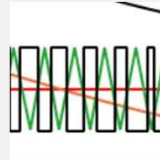
Trial 1 Successes



Motor Torque within spec



Secure connection between the wheel and motor shaft



Improved pulley system stability



Lifting effect mitigated

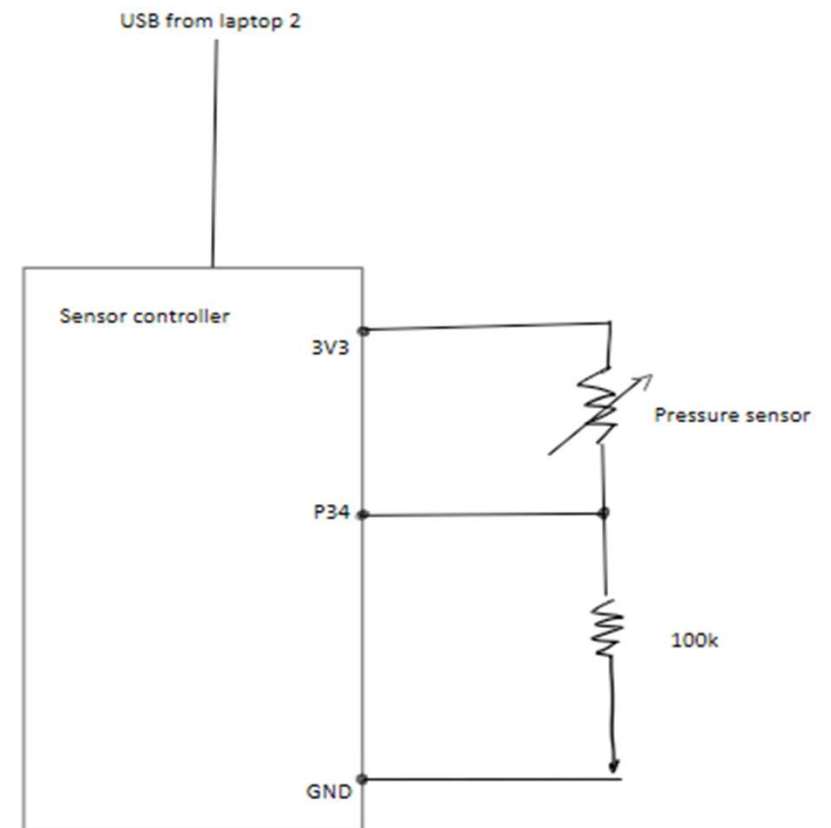
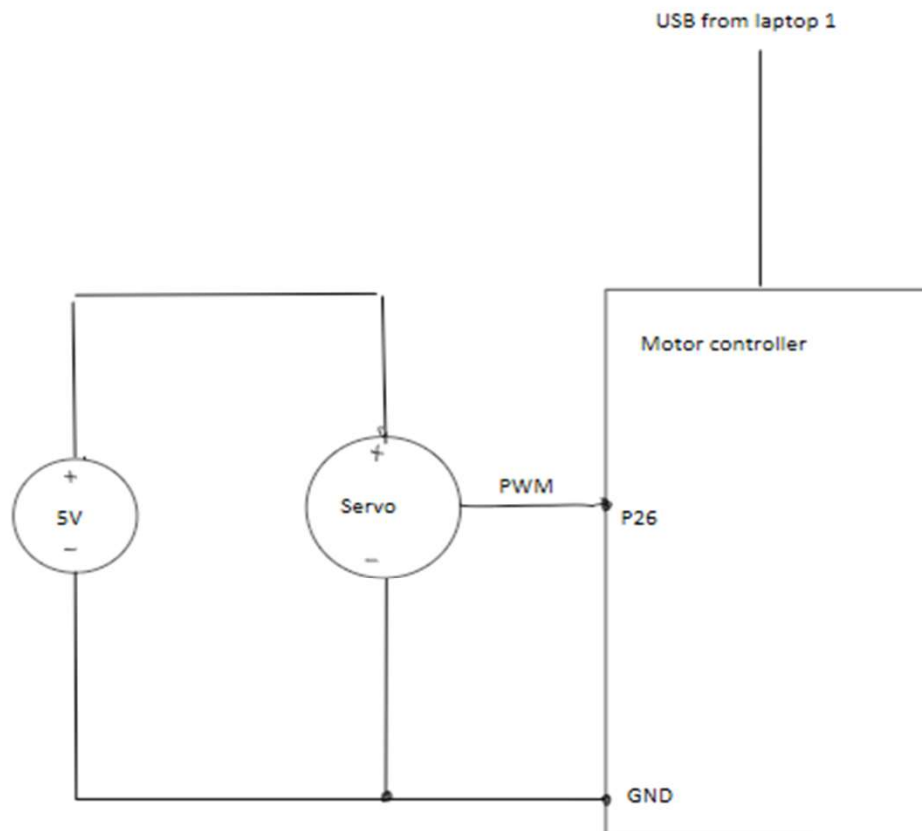
[Visual: ESP32 board image]

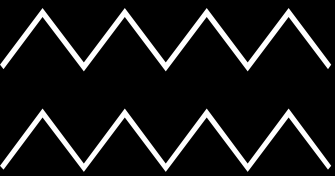


Motor and Structural Testing (2nd trial)

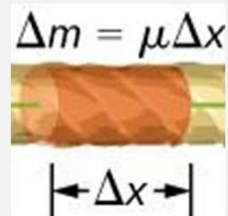


- Bolt/nut adjustment system for fine control of string tension
- Pulley system was redesigned with greater clearances to reduce friction.
- Resistive pressure sensors used in mouthpiece implemented.

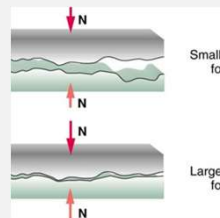




Trial 3 Successes



Fine control of spring tension.



Reduced friction in pulley system.



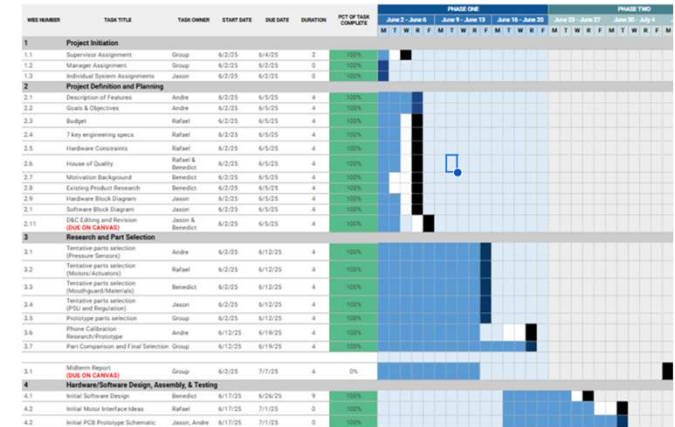
Pressure sensor functionality confirmed.



Comment	Description	Designator	Footprint	LibRef	Quantity	Manufacturer Part Number	Cost
MP2338GTL	Battery to 5V Buck				10		\$11.30
MS621FE-FL11E	MS Lihium Rechargeable Battery	Lion Battery?	SEIK-MS621FE-FL11E-2_V	CMP-00029-00000-1	2		\$13.29
ESP32-WROOM-32D	WIFI MODULE 32MBITS SPI FLASH	PeripheralController	FP-ESP32-WROOM-32D-MFG	CMP-194065-000003	1	ESP32-WROOM-32D	\$5.38
Boot Select button	Push Button Normally Opened	S1		Push Button NO	1		\$1.94
CP2102-GMR	Single-Chip USB to URAT Bridge	U?	SLAB-QFN-28-3125x3125TP_M	CMP-2000-04924-1	1		\$8.88
LP2966IMM-1833/NOPB	Dual 150mA Ultra Low-Dropout Regulator	U?	MUA08A_N	CMP-0062-02215-2	10		\$15.40
BQ25887	Battery Manager				2		\$10.00
18650 Battery Case					2		\$9.99
18650 Batteries					2		\$3.00
Satin String Spool					1		\$6.99
Servo Pulley					20		\$3.99
MG996R	Servo Motor				4		\$18.00
RP-S5	Resistive Pressure Sensors				2		\$9.50
Mouthguard					20		\$8.99
						Total Budget	\$122.66

Gantt Chart

Work Distributions



- While there is a table of work distributions. all members are actively communicating and working on every aspect of development.
- This allows all members to learn about topics we are weak in and teach others about topics within our strong areas.
- The Gantt chart is a more accurate depiction of our work distributions.

Work Distributions

Name	Primary Tasks	Secondary Tasks
Jason	Manager, Power	PCB
Andre	PCB Design	Coding
Rafael	Motors	PCB
Benedict	Coding	PCB



Progress and Plan for Completion

We have ordered our PCBs to evaluate the system's abilities to meet requirements, before implementing our stretch goals.

