



UNIVERSITY OF
ILLINOIS
URBANA-CHAMPAIGN

Forma the Smart-Clip

Electrical & Computer Engineering

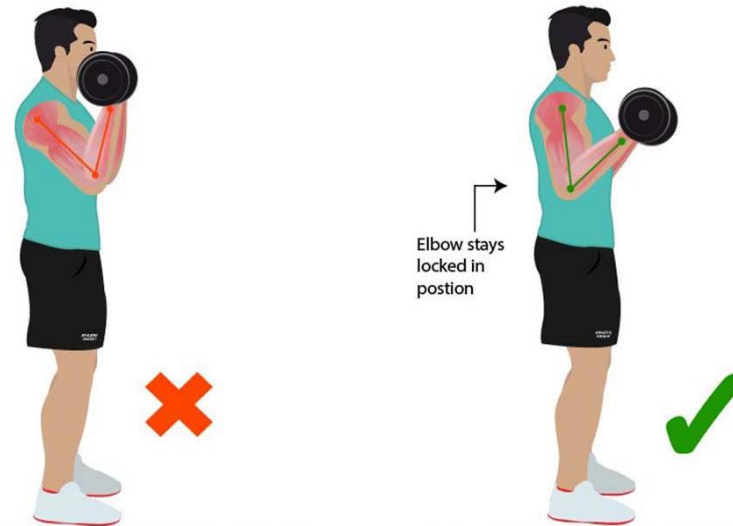
ECE 445 Senior Design, Spring 2026

Team #82: Sree Akkina, Bhanuprakash Kunam, Ishank Pujari

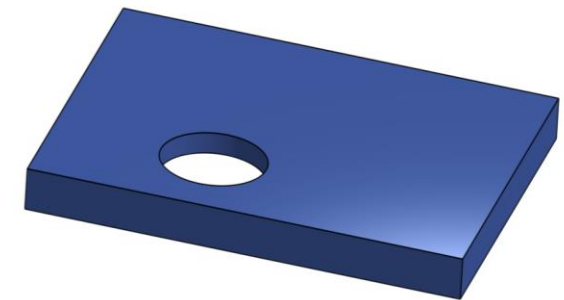
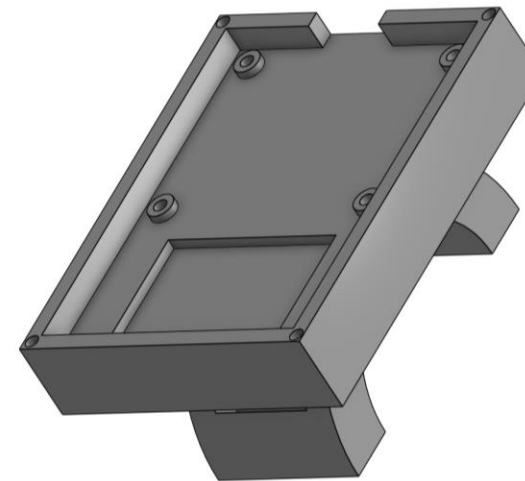
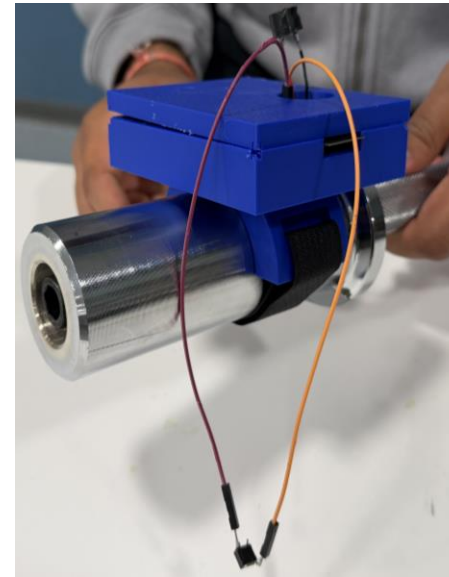
TA: Ko-Po Jen

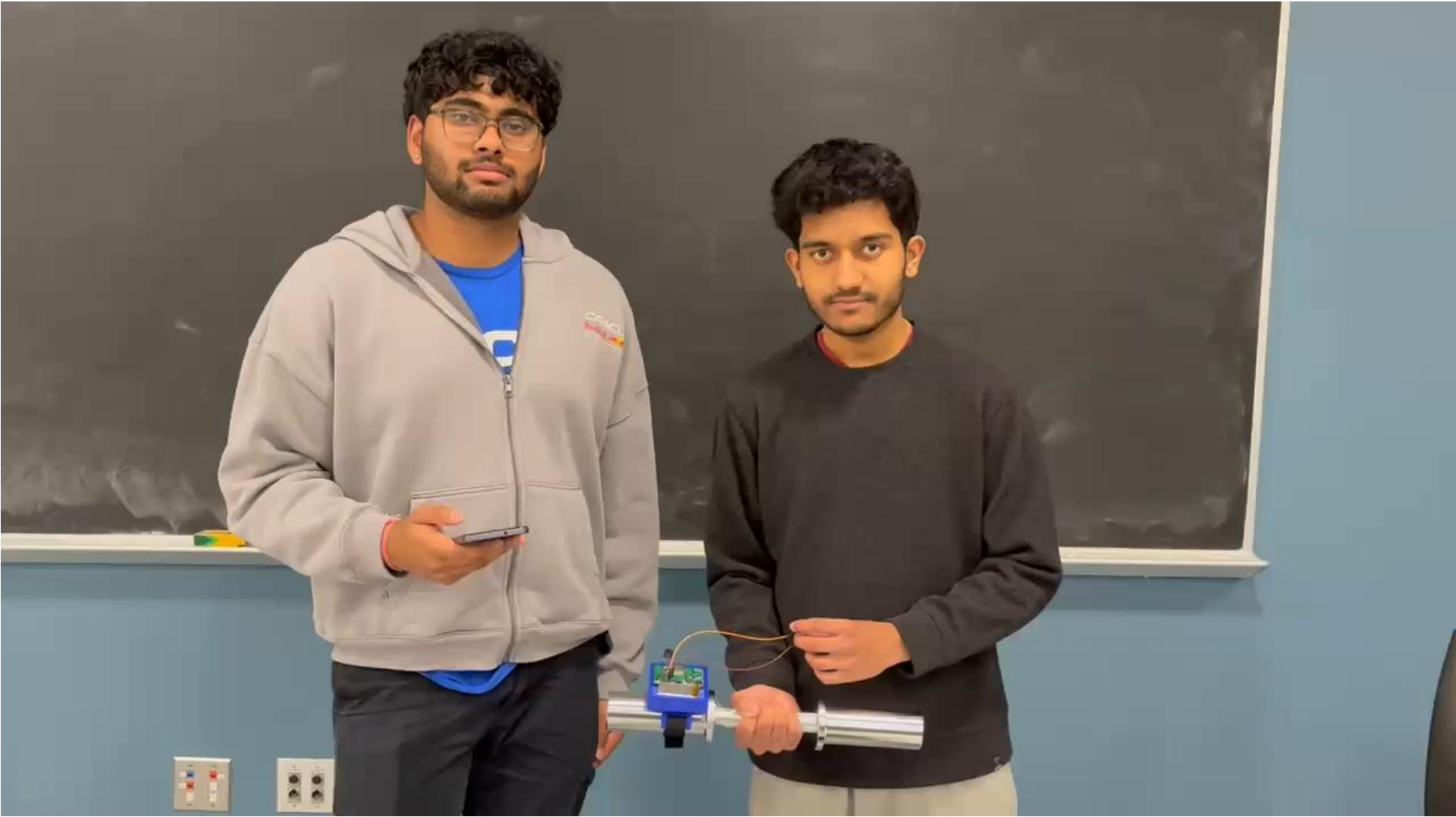
- **Bhanu Kunam**
 - Designed the PCB integrating the ESP32-S3, IMU sensor, and battery
 - Validated system performance through SPICE transient analysis of the buzzer subsystem and tested the hardware to ensure stable power levels and reliable data communication between all sensors
- **Sree Akkina**
 - Designed the PCB housing to firmly mount the electronics and battery onto the barbell.
 - Developed firmware, including module communication and helping with algorithm.
 - Soldered components onto PCB and handled testing subsystem connections.
- **Ishank Pujari**
 - Built the Android mobile app using React Native.
 - Handled all Bluetooth communication between the app and the device.
 - Helped with the rep detection and form grading algorithm on the ESP32.

- Lifters who train alone have no way to know if their form is breaking down mid set
- Looking at a phone or mirror mid lift is unsafe and breaks your focus
- Hiring a personal trainer is expensive and not always available
- No easy way to get real-time feedback on their form

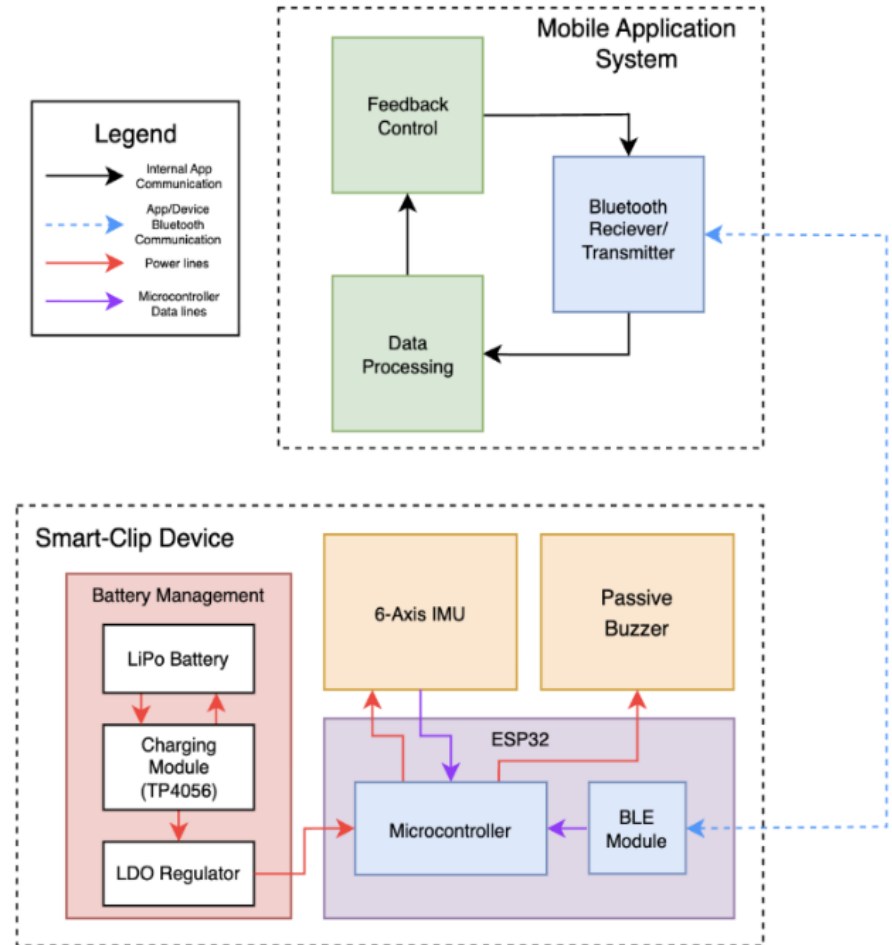


- Small device that clips onto a barbell.
- Uses ESP32-S3 microcontroller and MPU-6050 IMU to track motion.
- Provides real-time audio feedback through a piezo buzzer.
- Streams rep count and wobble data to a mobile app over BLE.
- 3D-printed housing assembly designed for a precision fit onto barbell, eliminating vibrations.

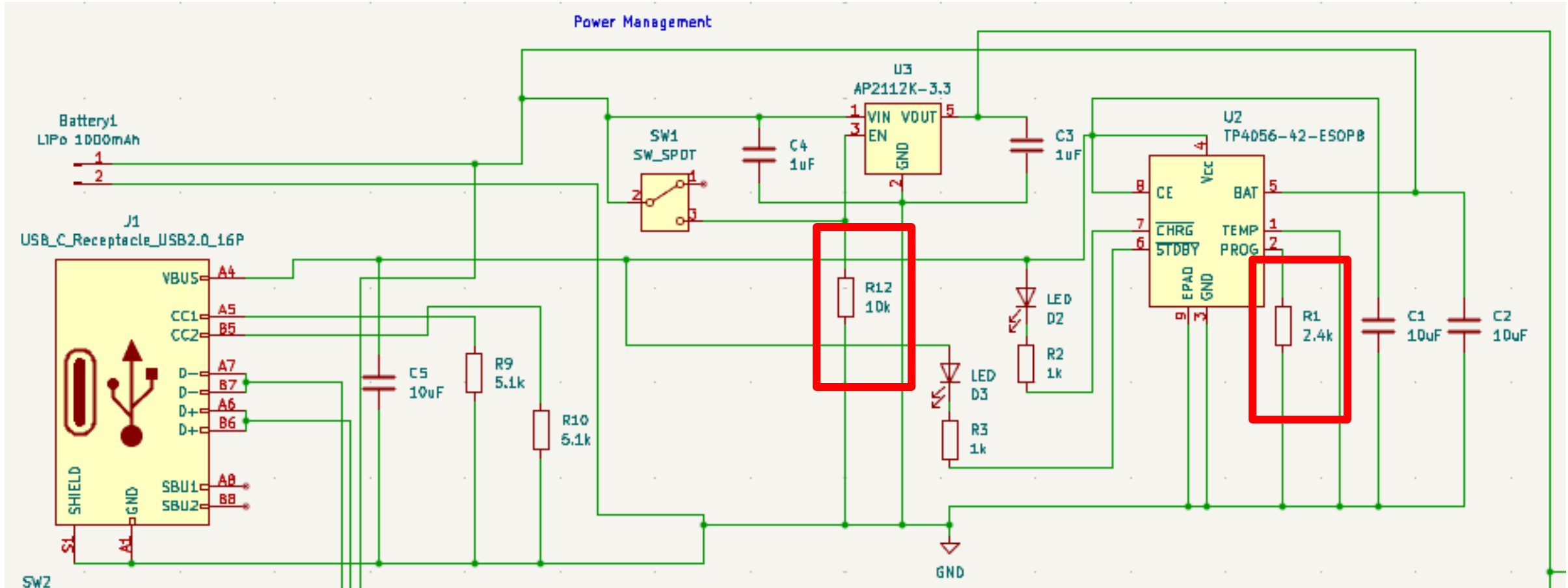




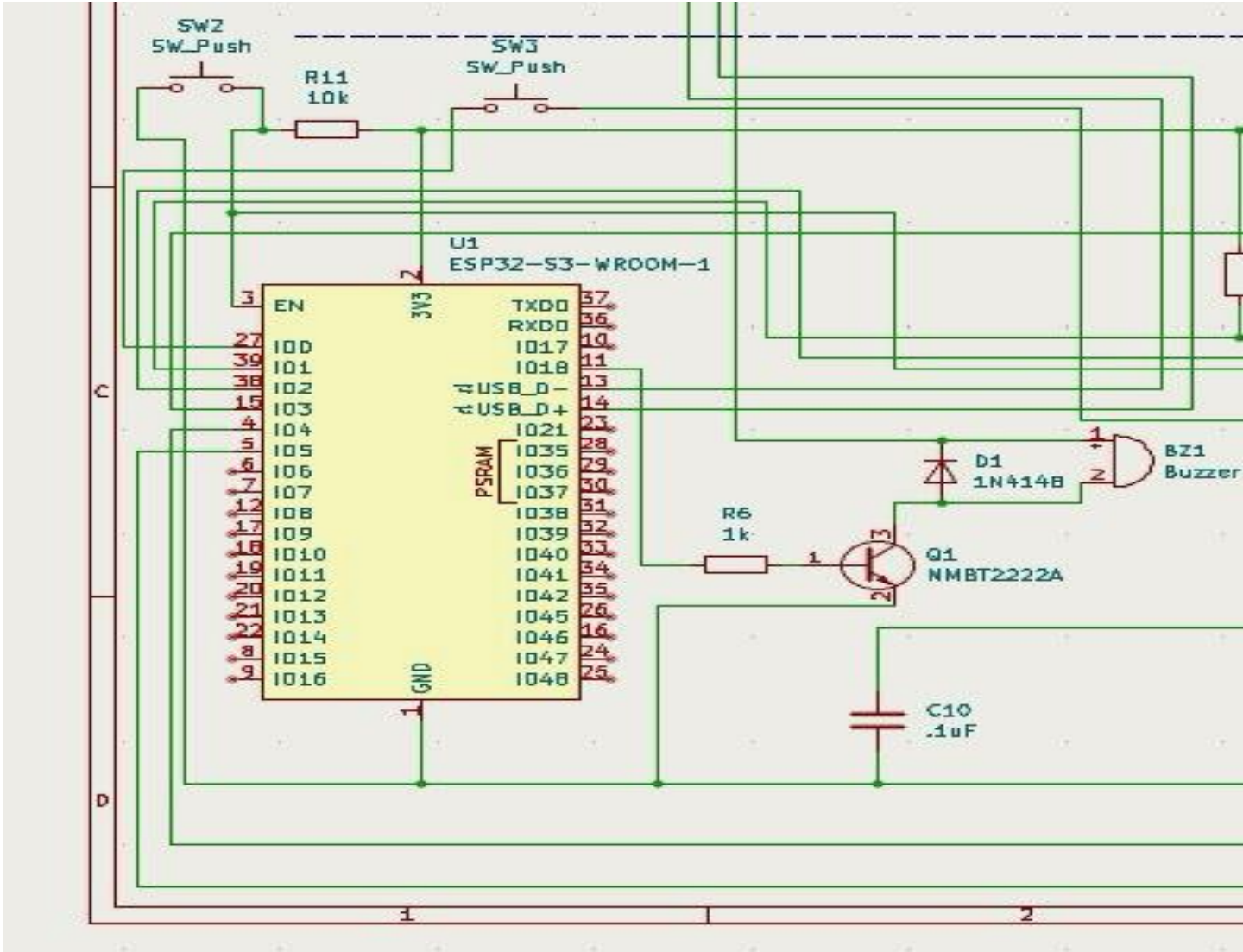
- **Processing Subsystem:**
 - ESP32-S3-WROOM-1 Microcontroller
- **Sensing Array:**
 - MPU-6050 via I2C for acceleration and position tracking.
- **Power Subsystem:**
 - TP4056 charging chip
 - AP2112K 3.3V LDO regulation
 - 1000 mAh LiPo battery
- **External Devices:**
 - PWM driven piezoelectric buzzer



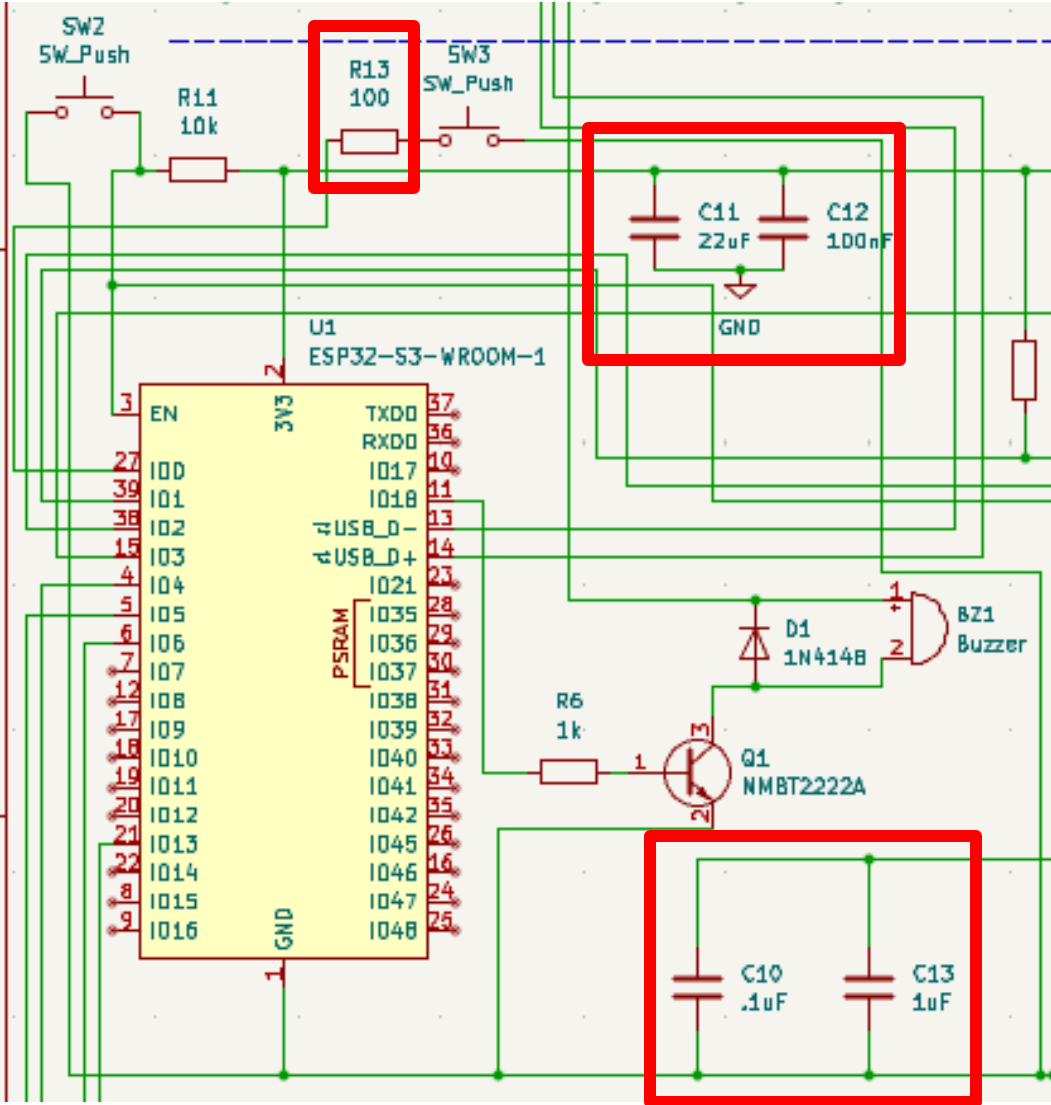
Original Design & What Changed



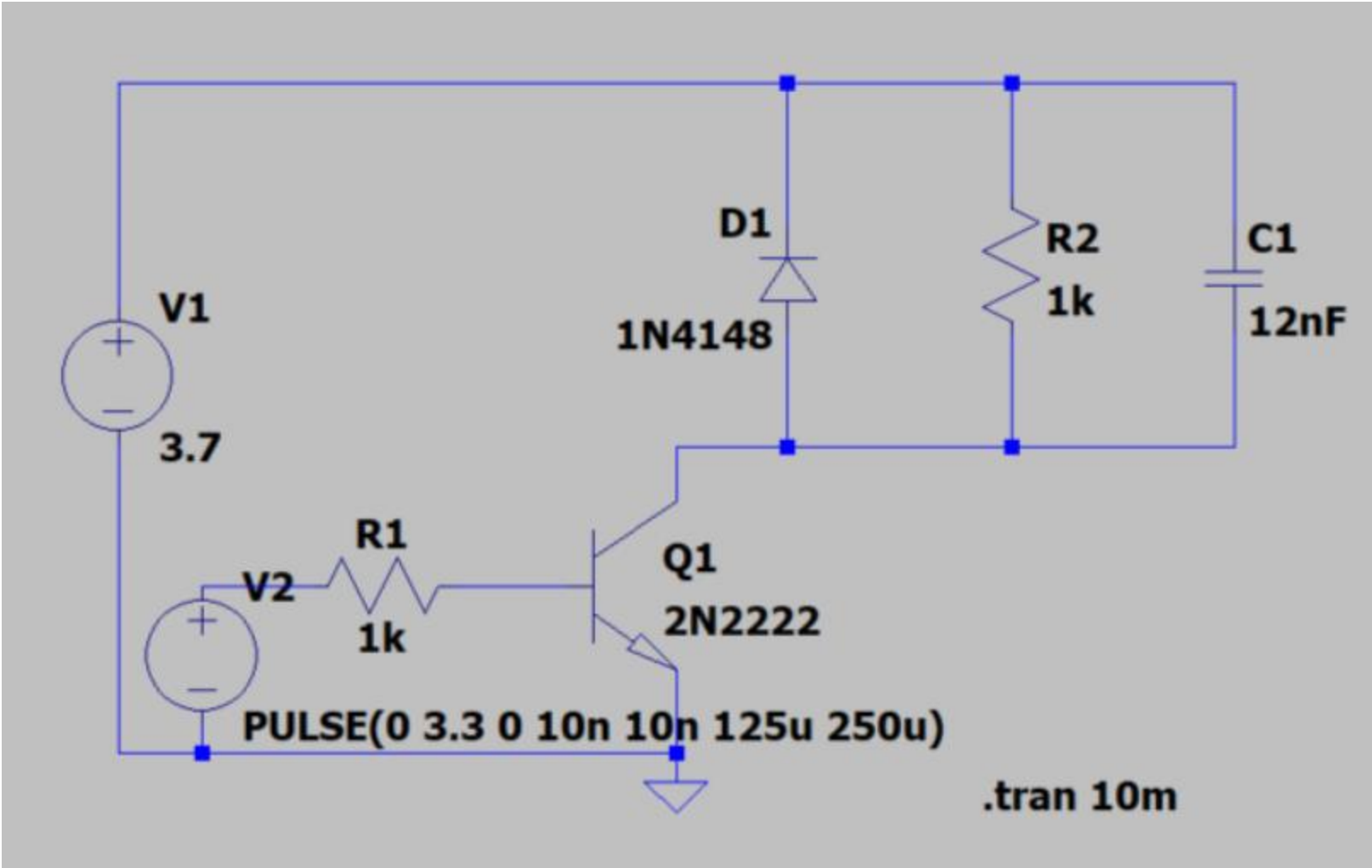
Original Design & What Changed



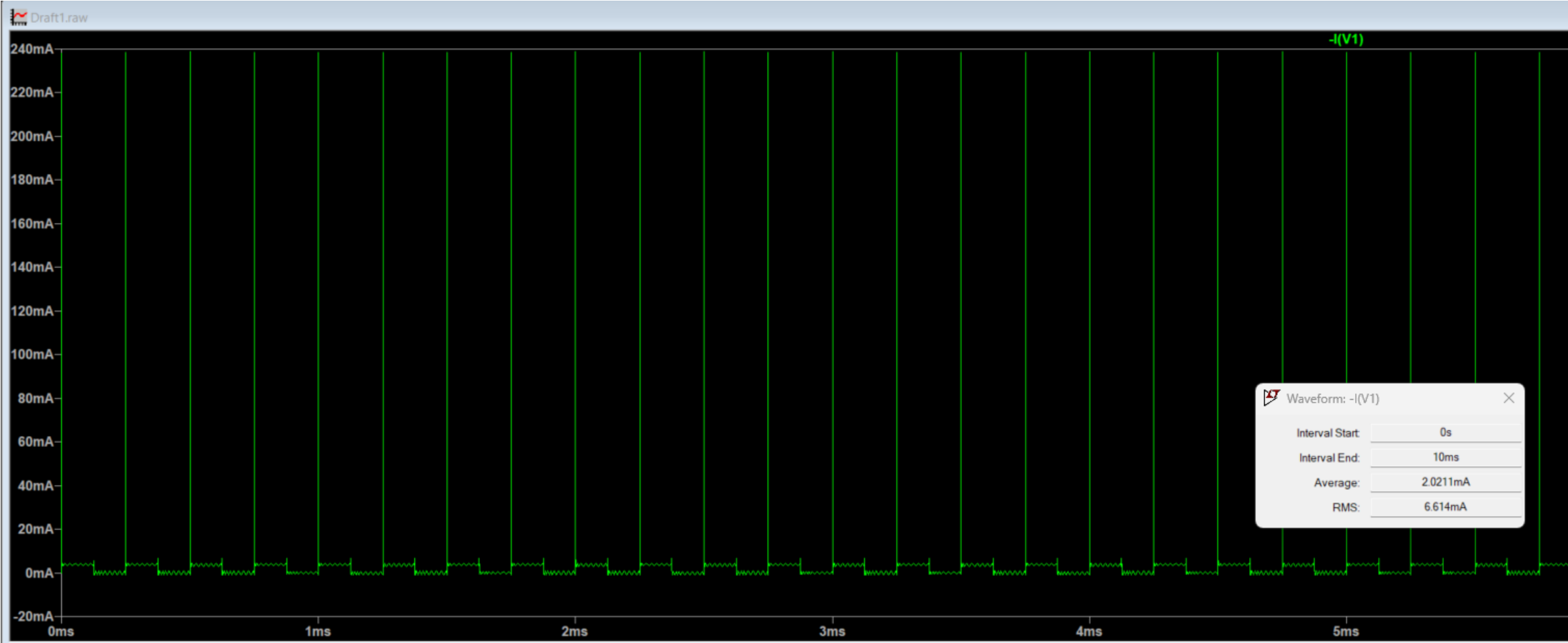
Original Design & What Changed



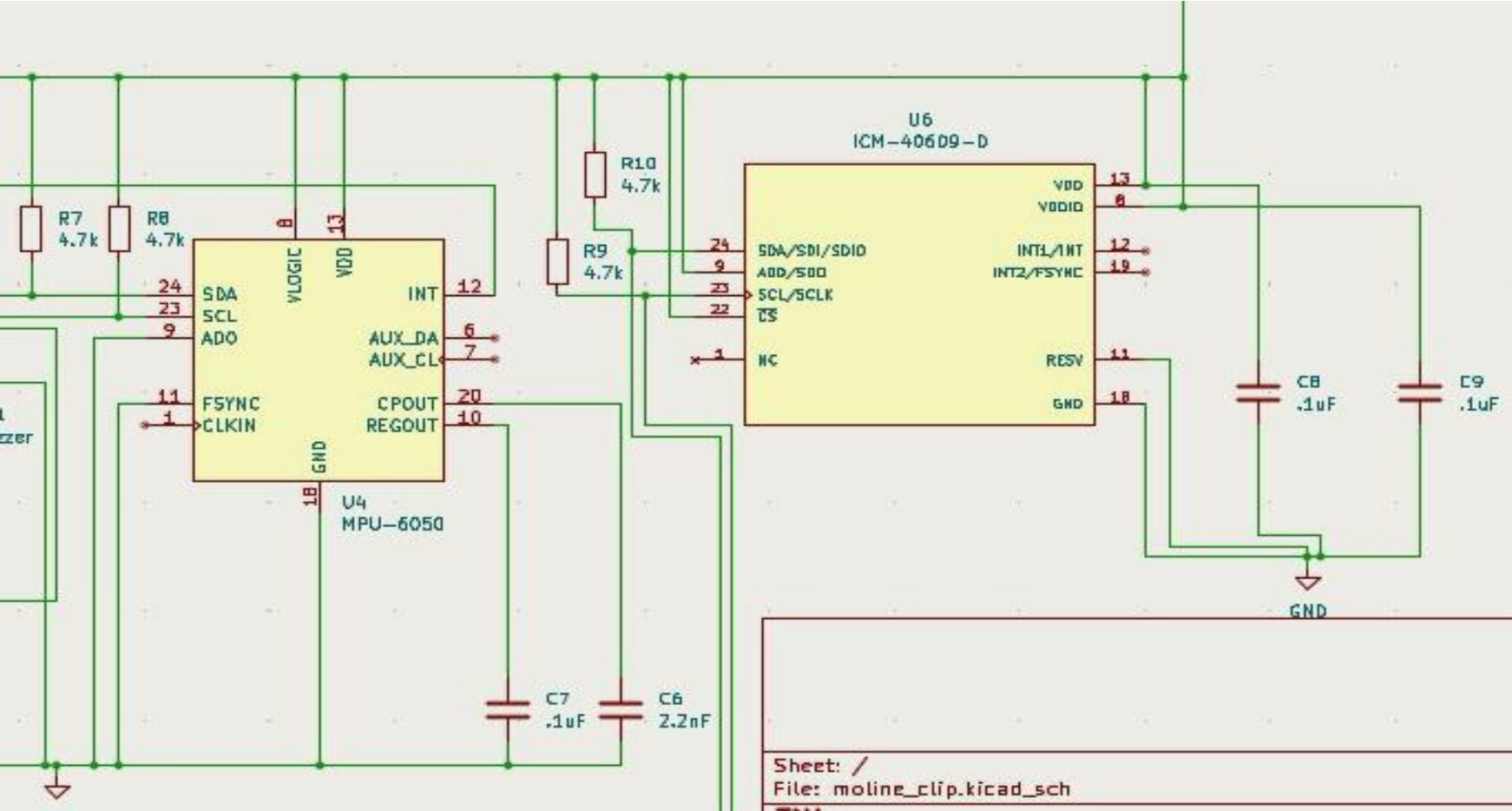
Original Design & What Changed



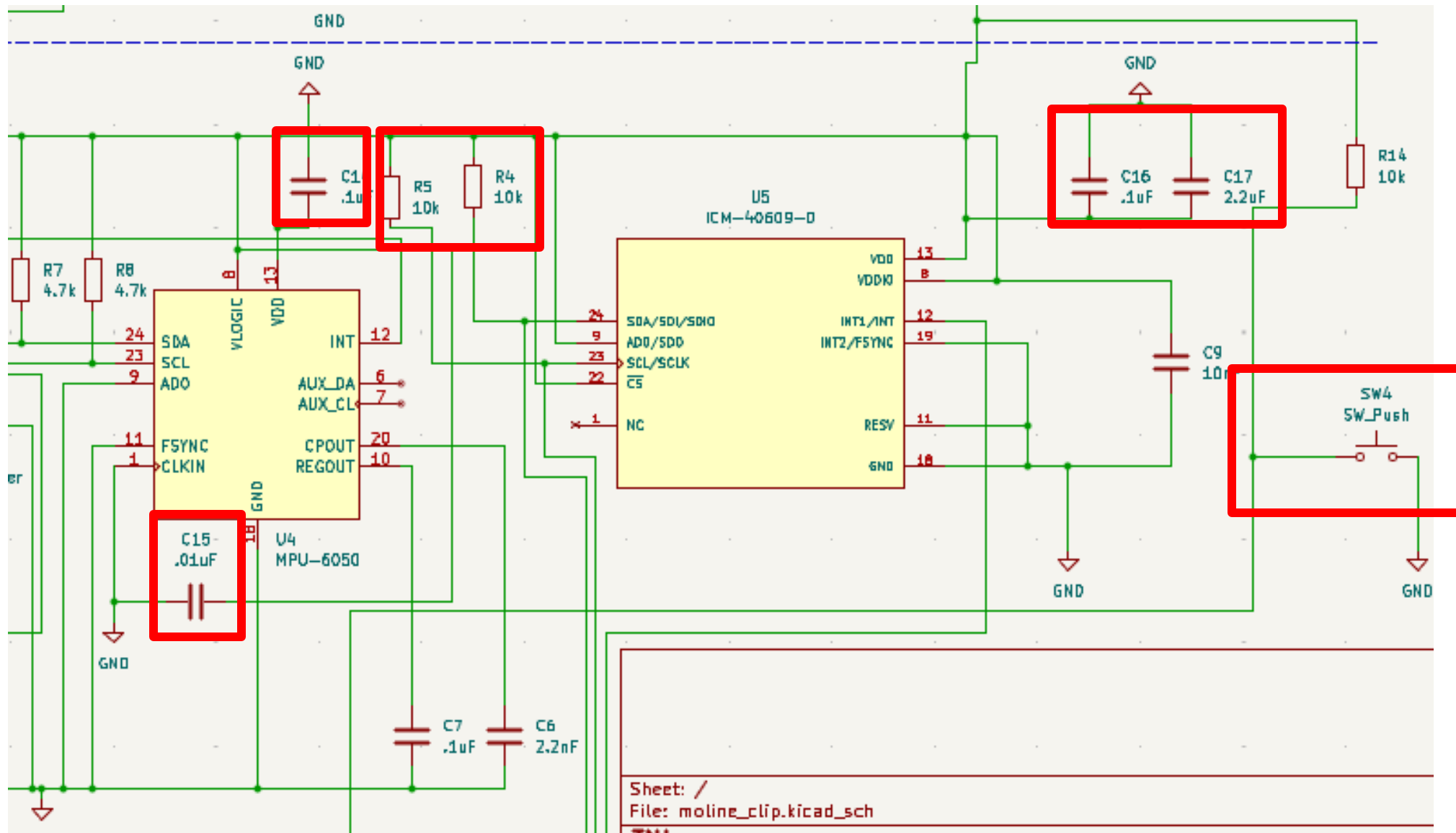
Original Design & What Changed



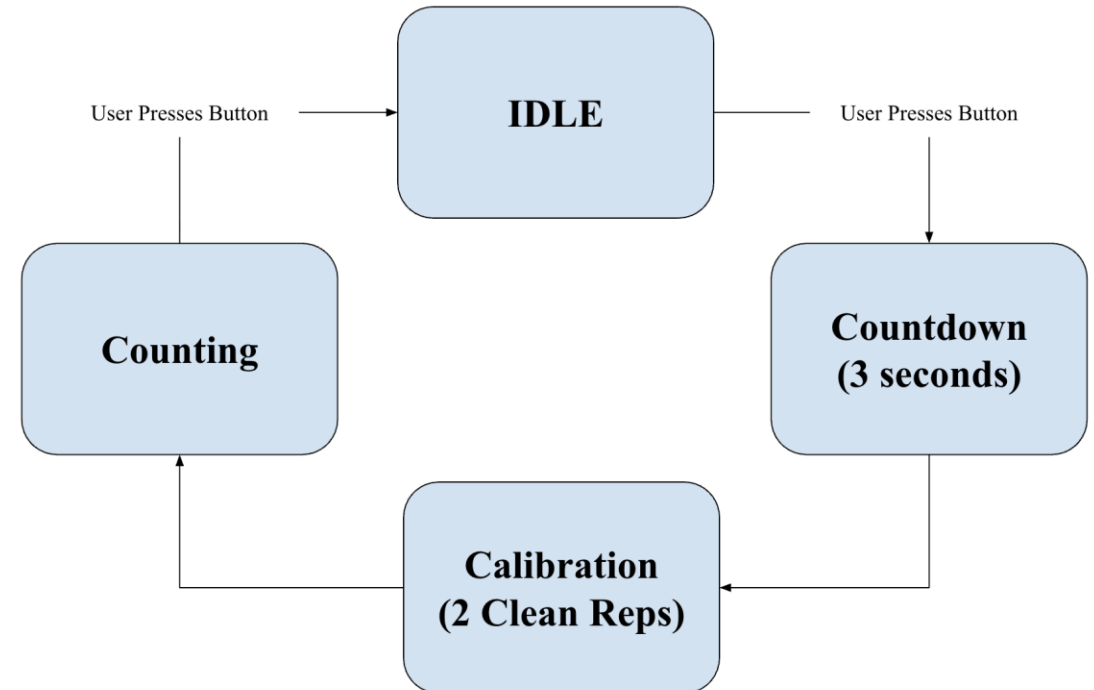
Original Design & What Changed



Original Design & What Changed



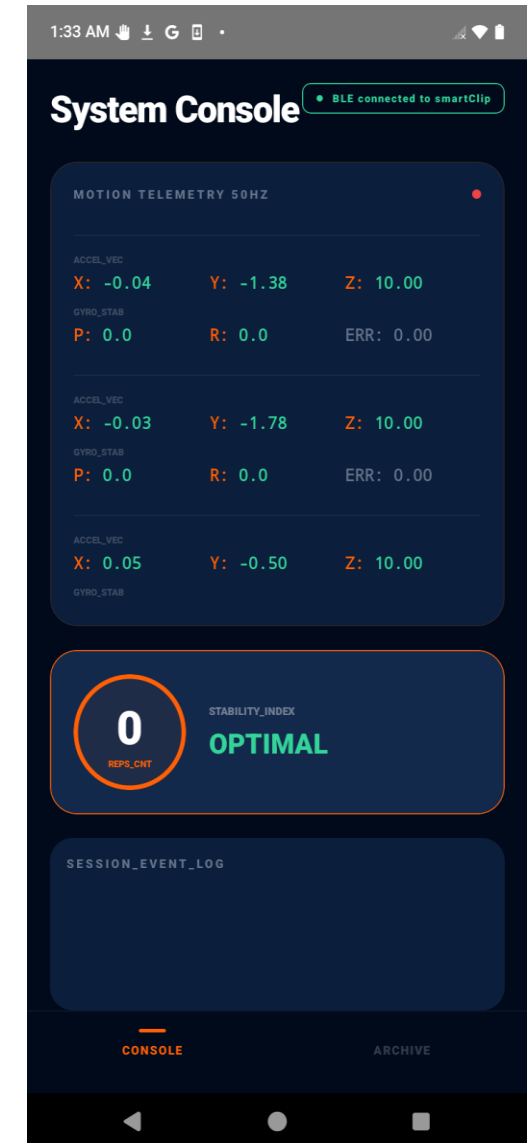
- **Hardware Integration**
 - Integrated control for 6-axis MPU6050, piezo buzzer feedback, and status LEDs (used for breadboard testing).
- **State Machine**
 - Idle: BLE standby mode and waiting for user button trigger.
 - Countdown: audio cues signal user to get into starting position.
 - Calibration: captures two clean reps to establish a baseline form template.
 - Counting: real-time monitoring and active rep analysis.



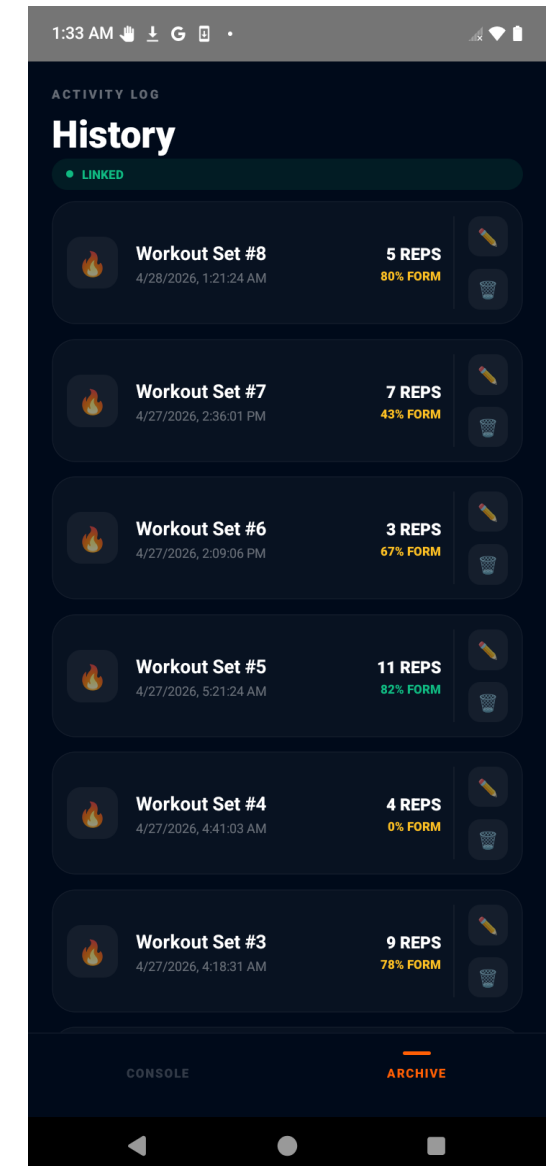
- **Form Template**
 - Averaged Baseline: combines data from two calibration curls (done with no additional weight) to create a template form signature.
 - Euclidean Distance: calculates the real-time deviation from the template across acceleration and gyroscope magnitudes.
- **Rep Detection Logic**
 - Threshold triggering: reps are started when the baseline deviation exceeds 3.0 m/s/s and each rep gets counted when the bar returns to the resting threshold.
- **Bad Form Detection**
 - Outside template form warning: Flags the reps if the deviation from the template exceeds a set distance.
 - Wobble Detection: Flags the reps if there is a 15-degree or higher tilt (calculated based on accelerometer values on x,y,z axes).



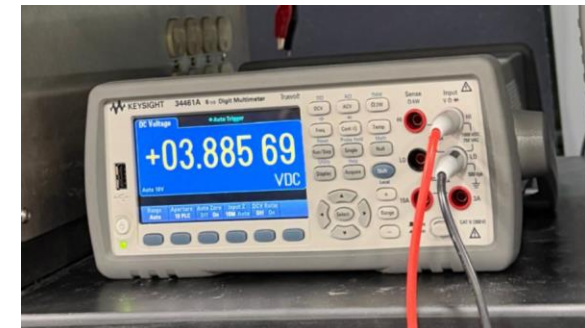
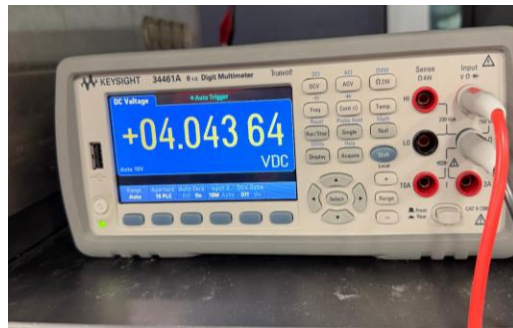
- **Architecture**
 - The app is built for Android using React Native
 - The ESP32 handles all the sensor data processing and form grading, while the app handles all the display, data logging, and user experience
 - The app uses useState and useEffect to show live data on screen
 - Everything works without wifi
- **Auto Connection**
 - Used the react-native-ble-manager library to handle Bluetooth
 - On the ESP32 side, we use the BLEDevice library to set up the device as a BLE server and make it discoverable.
 - The ESP32 advertises itself with the name smart-clip82 and its service information.
 - The app scans for our device and connects to smart-clip82 automatically when its nearby
 - The user never has to go into a pairing menu



- **Two Bluetooth Characteristics**
 - We split the data into two separate streams to avoid overloading the connection
 - The first stream sends live IMU sensor data 50 times per second, so the app can show real-time movement during each rep.
 - The second stream sends rep and set results only when a rep is completed, including the rep count and whether the form was good or bad.
- **Storing Workout Data**
 - We do not use a cloud database
 - The app takes the Bluetooth data and turns it into JavaScript objects directly
 - Rep counts, form pass/fail results, and timestamps are saved to the phone using AsyncStorage
 - The user can see all their past workouts in the History tab



Subsystem Requirements	Verification
Battery Charging Interface: Provides 4.2V (max) for charging and accepts 3.0V to 4.2V for discharging.	<ul style="list-style-type: none">• Ensure battery is charging.• Observe if the device operates under the correct voltage range.
Regulated Output Interface: Supplies a continuous 3.3V to the ESP32 with a maximum current capacity of 600mA.	<ul style="list-style-type: none">• Confirm with a multimeter that the output voltage is at a stable 3.3V across the current range.



Subsystem Requirements	Verification
Power Interface: Accepts a regulated 3.3V input and must operate within a current draw range of 80mA (idle) to 400mA (peak BLE transmission).	<ul style="list-style-type: none">Verify through datasheet
Wireless Interface (BLE): Must support Bluetooth Low Energy 4.2/5.0 protocols to maintain a stable link with a mobile device.	<ul style="list-style-type: none">Perform a simulated ten to twenty reset and confirm that all the recorded metrics transmit successfully to the app without dropping the connection.

Table 6-5. Current Consumption for Bluetooth LE in Active Mode

Work Mode	RF Condition	Description	Peak (mA)
Active (RF working)	TX	Bluetooth LE @ 20.0 dBm	344
		Bluetooth LE @ 9.0 dBm	202
		Bluetooth LE @ 0 dBm	187
		Bluetooth LE @ -15.0 dBm	119
	RX	Bluetooth LE	93

Subsystem Requirements	Verification
<p>Minimum Sampling Rate: The sensor must be configured to an output data rate of at least 50 Hz to capture high frequency “shakes” common in muscle fatigue.</p>	<ul style="list-style-type: none">Utilize a script to record the timestamp every time it receives new data from the IMU.Let the system run for 10 seconds and review the timestamps and confirm at least 500 distinct data points in the window.

Subject 1 183 cm Rep 1 512 samples @ 50Hz												
--- CSVSTART ---												
subject_id	height_cm	rep_num	sample_num	time_ms	ax_g	ay_g	az_g	gx_dps	gy_dps	gz_dps	accel_mag_g	tilt_angle_deg
1,183,1,0,203,-0.3232,1.0894,10.7524,0.5954,3.6031,0.8244,10.8122,6.0326												
1,183,1,1,222,-0.3328,1.1684,10.6949,-0.6412,3.4198,1.4809,10.7637,6.4805												
1,183,1,2,242,-0.3400,1.1684,10.7428,-2.0763,2.9618,2.0458,10.8115,6.4623												
1,183,1,3,262,-0.2729,1.1732,10.7595,-2.0458,2.5191,2.2901,10.8268,6.3875												
1,183,1,4,282,-0.2370,1.1779,10.7595,-1.9237,2.1221,2.1527,10.8264,6.3720												
1,183,1,5,302,-0.2729,1.2163,10.7356,-2.2748,1.8015,1.8168,10.8077,6.6229												
1,183,1,6,322,-0.2777,1.2186,10.7907,-1.7405,1.4504,1.5725,10.8628,6.6072												
1,183,1,7,342,-0.2610,1.2641,10.7188,-2.4580,1.0992,1.3282,10.7963,6.8667												
1,183,1,8,362,-0.2586,1.1708,10.8529,-2.6260,0.7328,0.9313,10.9189,6.3042												

Subsystem Requirements	Verification
<p>BLE Data Reception & Display: The mobile application must receive data from the Smart-Clip via Bluetooth Low Energy (BLE).</p>	<ul style="list-style-type: none">• Mobile application is actively paired with the Smart-Clip via BLE and a brief workout is performed.• Confirm that the app is receiving real-time metrics from the Smart-Clip by monitoring the data logs.
<p>Real-Time Display: The application must use backend scripts to analyze incoming IMU data to provide repetition counts and stability metrics.</p>	<ul style="list-style-type: none">• Connect mobile app to the Smart-Clip which is actively in motion.• Confirm that the interface updates with rep counts and stability ratings.



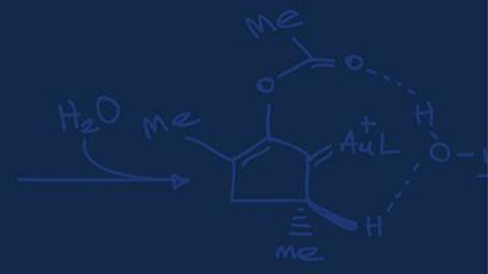
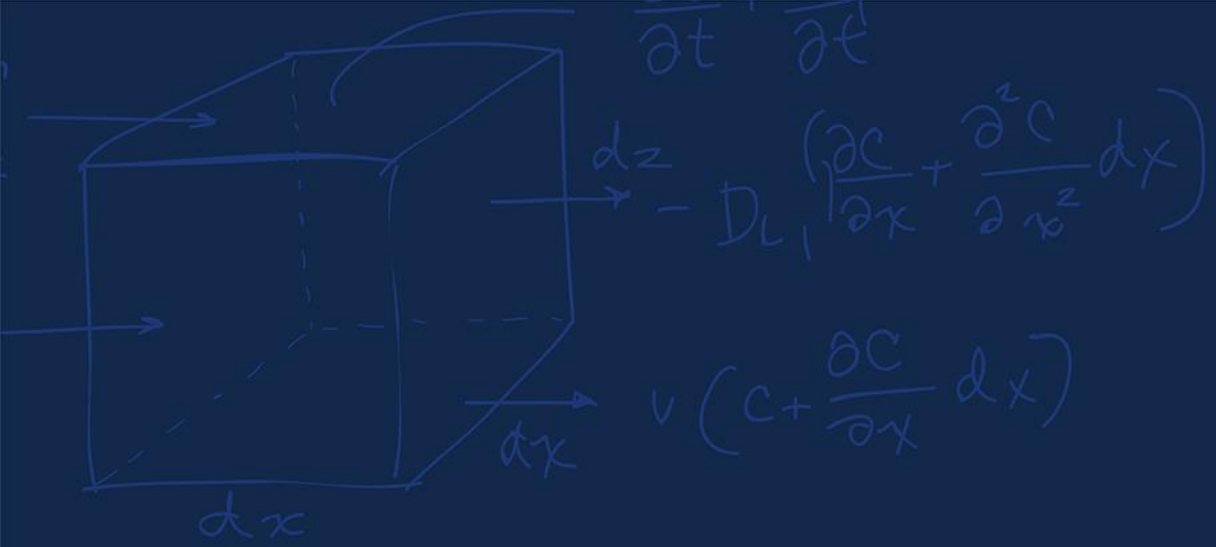
Successes	Challenges
<ul style="list-style-type: none">• Designing PCB to support all the subsystems.• Ensuring all the subsystems were tested individually and together.• Designing 3D-printed housing to precisely mount the PCB and battery onto barbell.• Integrating calibration-based form detection distinct to each set.• Device operating solely on battery power and holding stable connection with app via BLE.	<ul style="list-style-type: none">• Soldering components onto PCB (USB-C port, battery wires, and MPU6050).• Creating the form template model based on recorded reps from different people.• Debugging the PCB• Handling Bluetooth connection

Conclusion

- Designing a PCB and soldering components was a completely new experience for us and hands on work like that is something you can only truly learn by doing it yourself
- Sensor noise from the MPU6050 means you cannot trust raw readings alone, we had to spend time tuning our thresholds before rep detection became consistent.

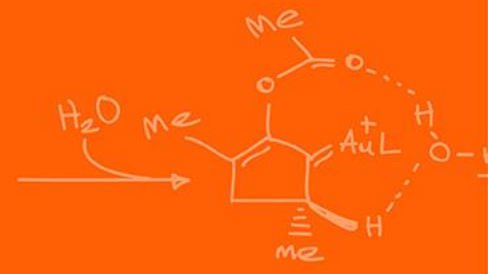
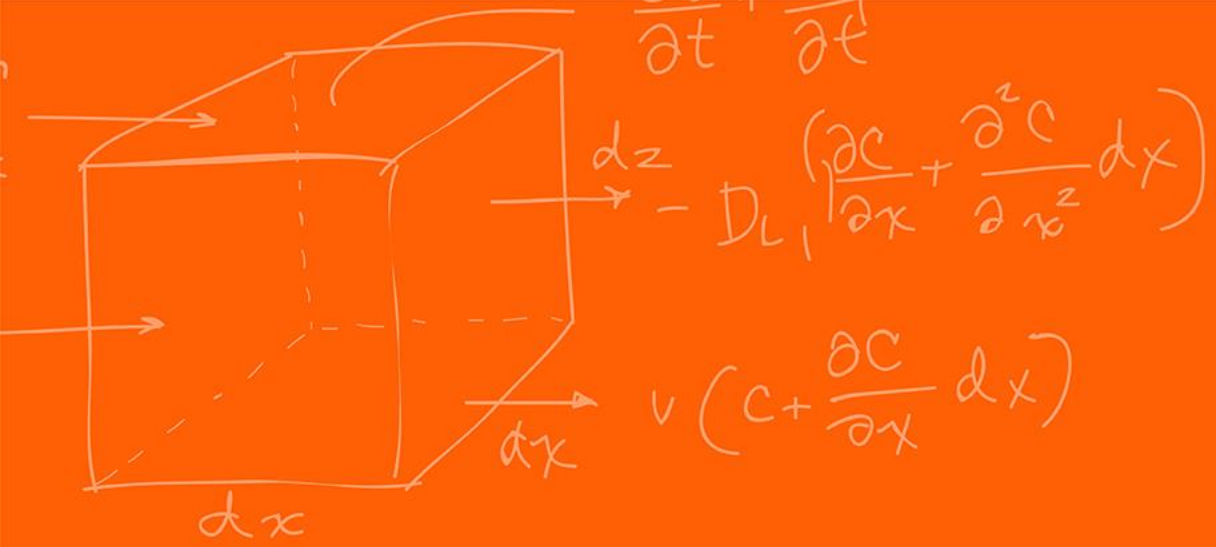
Future Steps

- Keep improving the algorithm and app to work with more exercises beyond just curls
- Iterate on the physical PCB layout and housing design to optimize the device's form factor for durability and precision.
- Explore cross-platform compatibility and cloud integration to allow users to track long-term progress and sync data across multiple devices.



Thank You! Any Questions?





The Grainger College of Engineering

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

