



EE 496 — Senior Capstone

Smart ACL Recovery Brace

Komal Kackar | Abir Bhatt | Anthony Chandra



Overview

ACL injuries are one of the **most common and challenging knee injuries** for active individuals. After surgical reconstruction, recovery **requires months of rehabilitation** focused on rebuilding strength, stability, and proper movement patterns. However, much of this process occurs outside of supervised physical therapy, **leaving patients** to perform exercises independently **without real-time guidance or feedback**. To address this gap, we **developed a smart ACL brace** that **tracks movement** and provides **real-time feedback**, helping users **improve form, build confidence**, and recover more effectively.



Market Research

Key Challenges

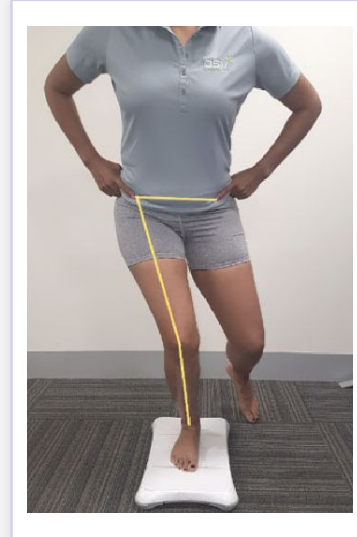
- Rehabilitation (12-18 months) occurs mostly unsupervised
- Patients have to relearn proper movement
- Improper form goes undetected (impacts recovery)
 - Uncertainty about performing exercises correctly
 - Uncertainty about proper weight distribution and movement

Unsafe Conditions

- What are unsafe joint angles/weight distributions?
 - Knee Flexion
 - Warning zone $> 90^\circ$
 - Danger zone $> 110^\circ$
 - Knee Valgus
 - Danger zone about $> 10-15^\circ$
 - Weight distribution
 - Danger zone 70-85% symmetry

Lack of Feedback

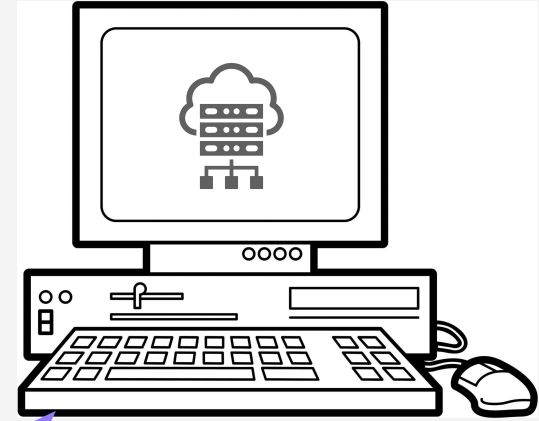
- Patients rely on subjective feedback like pain or discomfort to judge progress
- **Overall lack of confidence/trust in their own body**



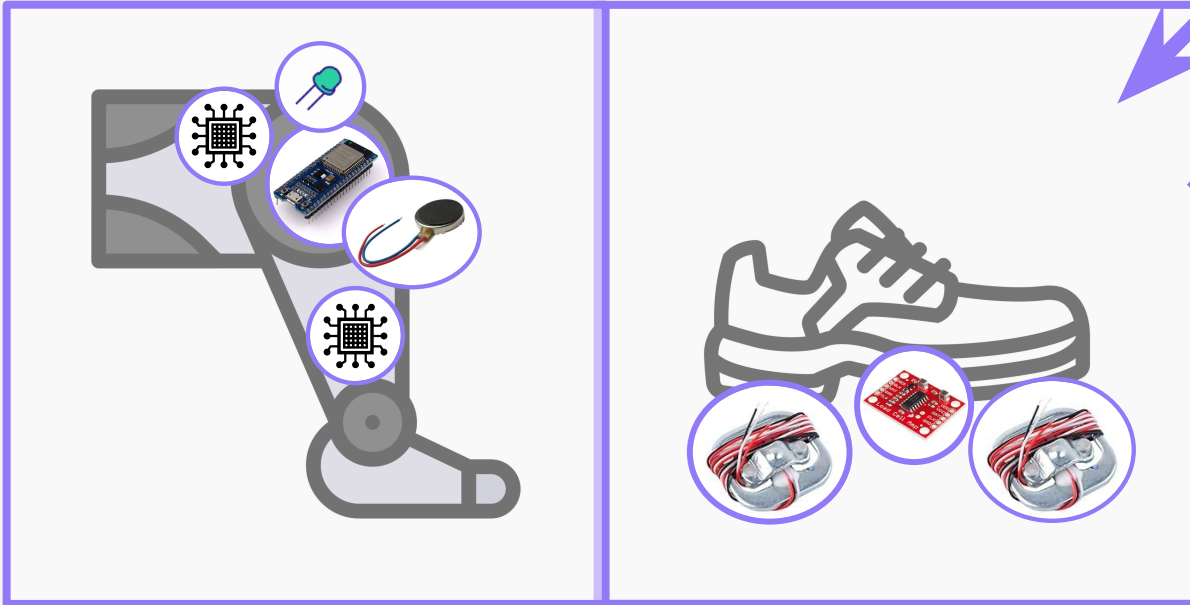
Overall Design

ESP32 & Knee Angles/Haptics

Load Cell



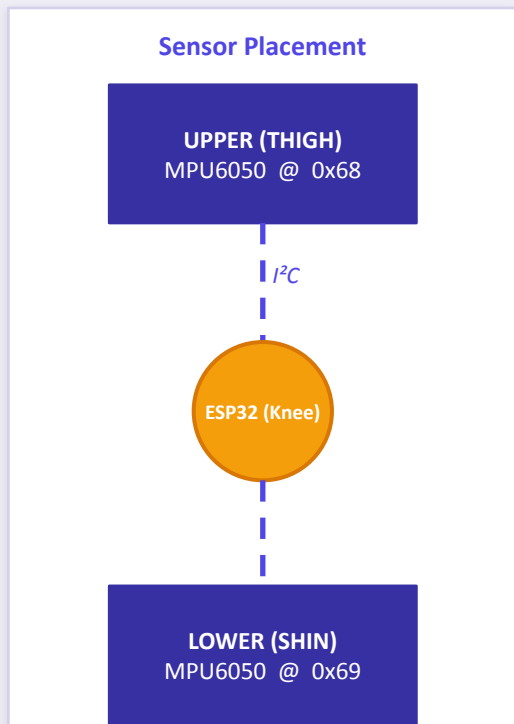
Web Server



IMU Integration — Sensor Architecture



Two MPU6050s sharing one I²C bus — upper (thigh) and lower (shin)



MPU6050 — 6-DoF IMU



3-axis gyroscope (± 2000 °/s) + 3-axis accelerometer (± 16 g). Communicates over 400 kHz I²C. One sensor strapped to thigh, one to shin.

One Bus, Two Sensors



AD0 pin selects the I²C address: upper IMU = 0x68 (AD0 tied LOW), lower IMU = 0x69 (AD0 tied HIGH). Both sensors share a single SDA/SCL wire pair.

Load Cell Bus Isolation



HX711 ADC uses dedicated GPIO pins (DT → GPIO 18, SCK → GPIO 19) — completely separate from I²C — eliminating bus contention and read-timing conflicts with the IMUs.



Knee Flexion & Valgus — Metrics

From raw IMU angles to clinically meaningful measurements — with coupling artifact correction

Knee Flexion (bend angle)

$$\text{flexion} = | \text{upper_pitch} - \text{lower_pitch} |$$

Reads 0° at natural standing. Alert triggers when > 90°.

Knee Valgus (lateral cave)

$$\text{valgus} = (\text{lower_roll} - \text{upper_roll}) - 0.17 \times \text{flexion}$$

Alert triggers when > ±10°.



Valgus Error Found During Testing

Bending the knee caused spurious valgus readings even when the leg was perfectly straight laterally. The lower IMU's roll axis picks up a gravity component as the shin tilts forward.

We plotted roll vs. flexion from and the CSV data showed a linear ~0.17°/° relationship. We fixed this by subtracting (0.17 x flexion) from the roll difference.



WebSocket Communication

Directly from ESP32 to browser (no cloud or server needed)



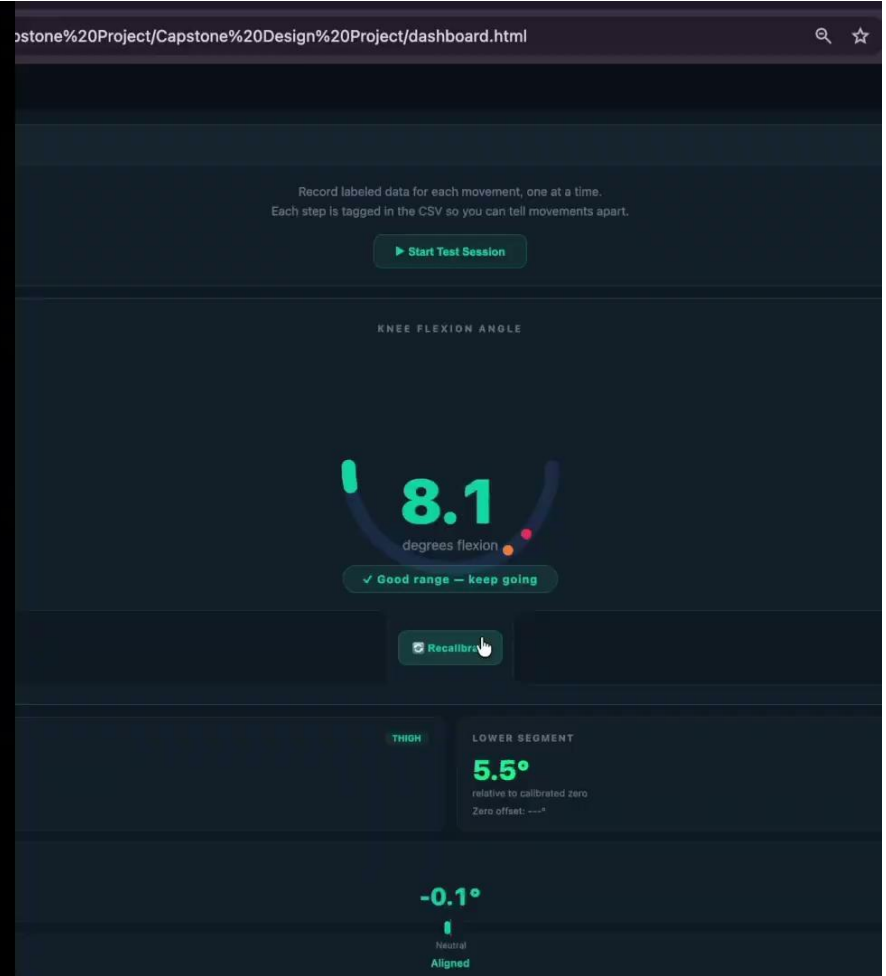
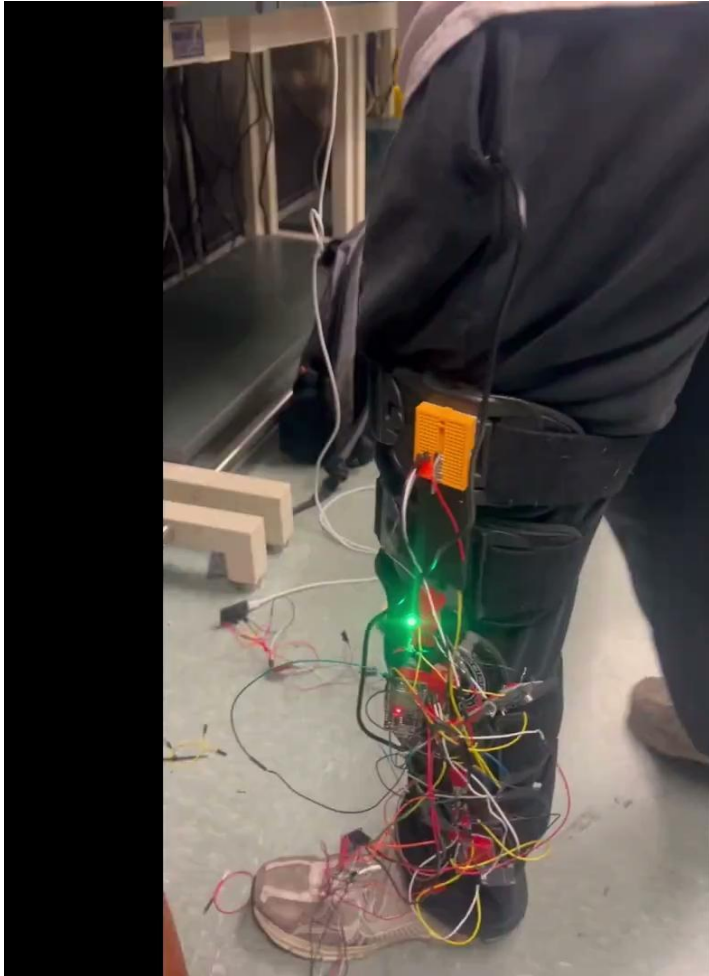
Why WebSocket?

- Chose over REST/MQTT because we needed fast, direct communication
- ESP32 runs as WiFi Access Point — Web UI can be accessed by any device on the same network
- WebSocket pushes at 20 Hz (50 ms) — a bit laggy right now (would try and fix in the future)
- Bidirectional: browser can send 'calibrate' or 'tare' commands back to ESP32

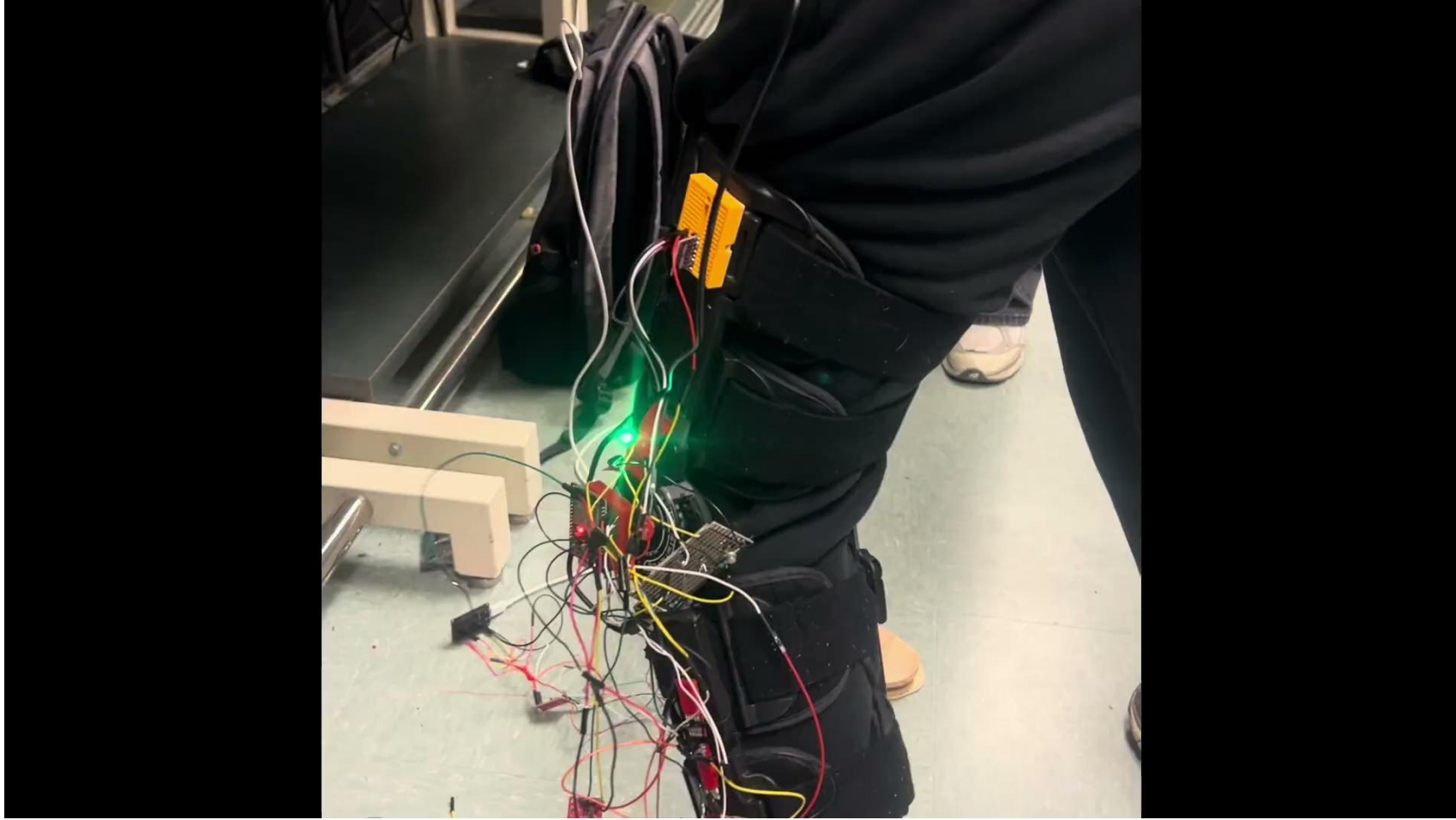
Example JSON Broadcast (every 50 ms)

```
{ "knee_angle": 42.3,  
  "valgus": -2.1,  
  "heel_kg": 28.4,  
  "toe_kg": 19.2,  
  "load_total": 47.6,  
  "alert": false,  
  "ts": 14832 }
```

Knee Flexion



Knee Valgus



Alerts & Session Score

Viewable in UI



Example Session Safety

97.4%

% of time in safe zone

Example Session Peak

88.2°

max flexion recorded

Example Alerts Logged

3

events this session

Session Safety Score

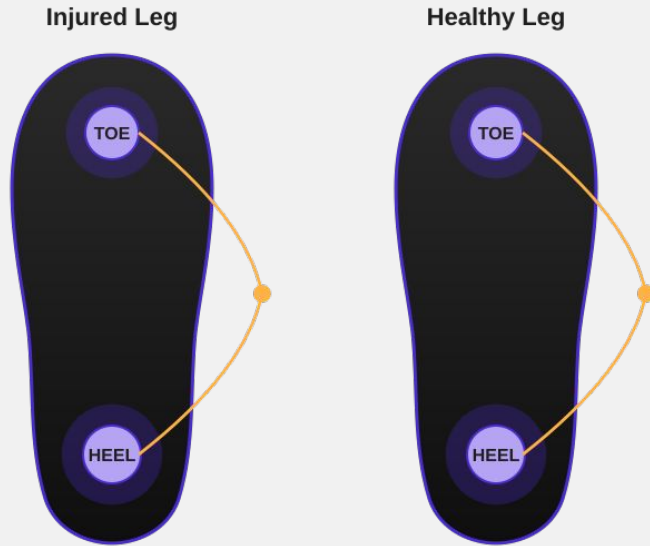
```
score = safe_packets / total_packets × 100%
```

- **≥ 95%**
Excellent form maintained throughout
- **80–95%**
Some limit exceedances — review alert log
- **< 80%**
Significant form breakdown — adjust exercise



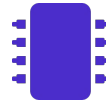
Load Sensing — Sole Design

Two load cells per sole — one at the heel, one at the toe



Heel + Toe split

Separates ground reaction force by foot region — flags toe-loading or heel-loading bias



HX711 24-bit ADC

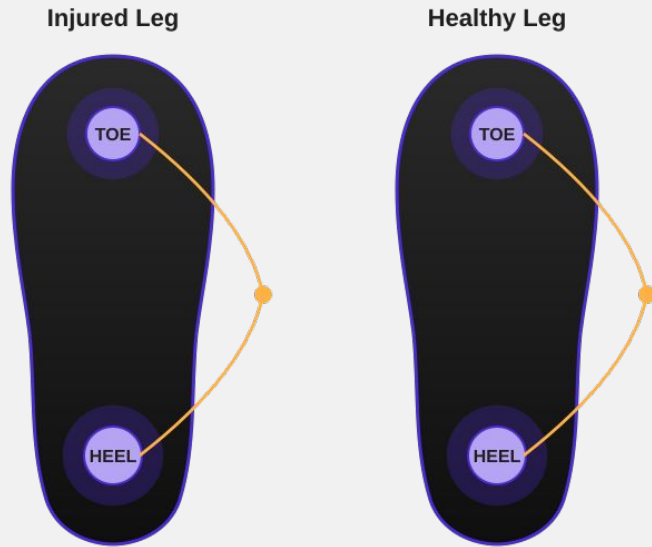
Amplifies the load cell bridge signal and feeds the ESP32 over GPIO. One HX711 reads both cells



Heel + Toe = Total

Per-region readings sum to total leg load — used downstream for cross-leg symmetry

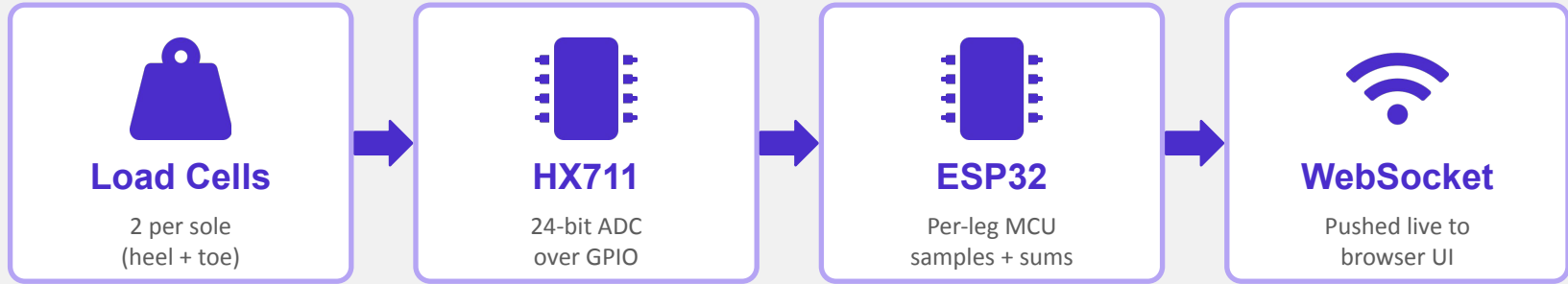
Load Sensing — Sole Design





Integration & Signal Path

How load data flows from the foot to the dashboard



Why this design

- 100 kg cells handle full body weight + dynamic gait loads
- Region-split (heel/toe) caught data a single insole cell would miss
- HX711 over GPIO — no I²C contention with the IMUs

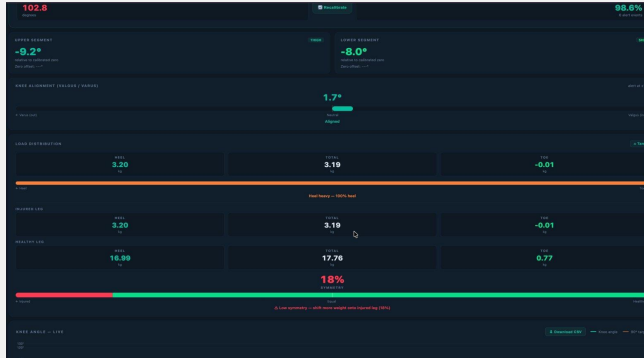
What I'd change

- Add a midfoot cell — 3 zones gives a cleaner gait signal
- Thinner plates or cushioning for better in-shoe comfort
- On-MCU low-pass filter to cut step-impact noise before transmit



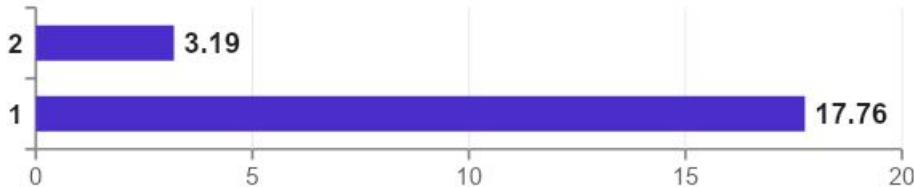
Load Symmetry — The Payoff

Comparing left vs right leg load shows the rehab discrepancy in real time



Our live dashboard (real run)

Live readings → 3.19 / 17.76 ≈ 18% symmetry (alert!)



18%

Symmetry score (min / max load × 100)



Threshold logic

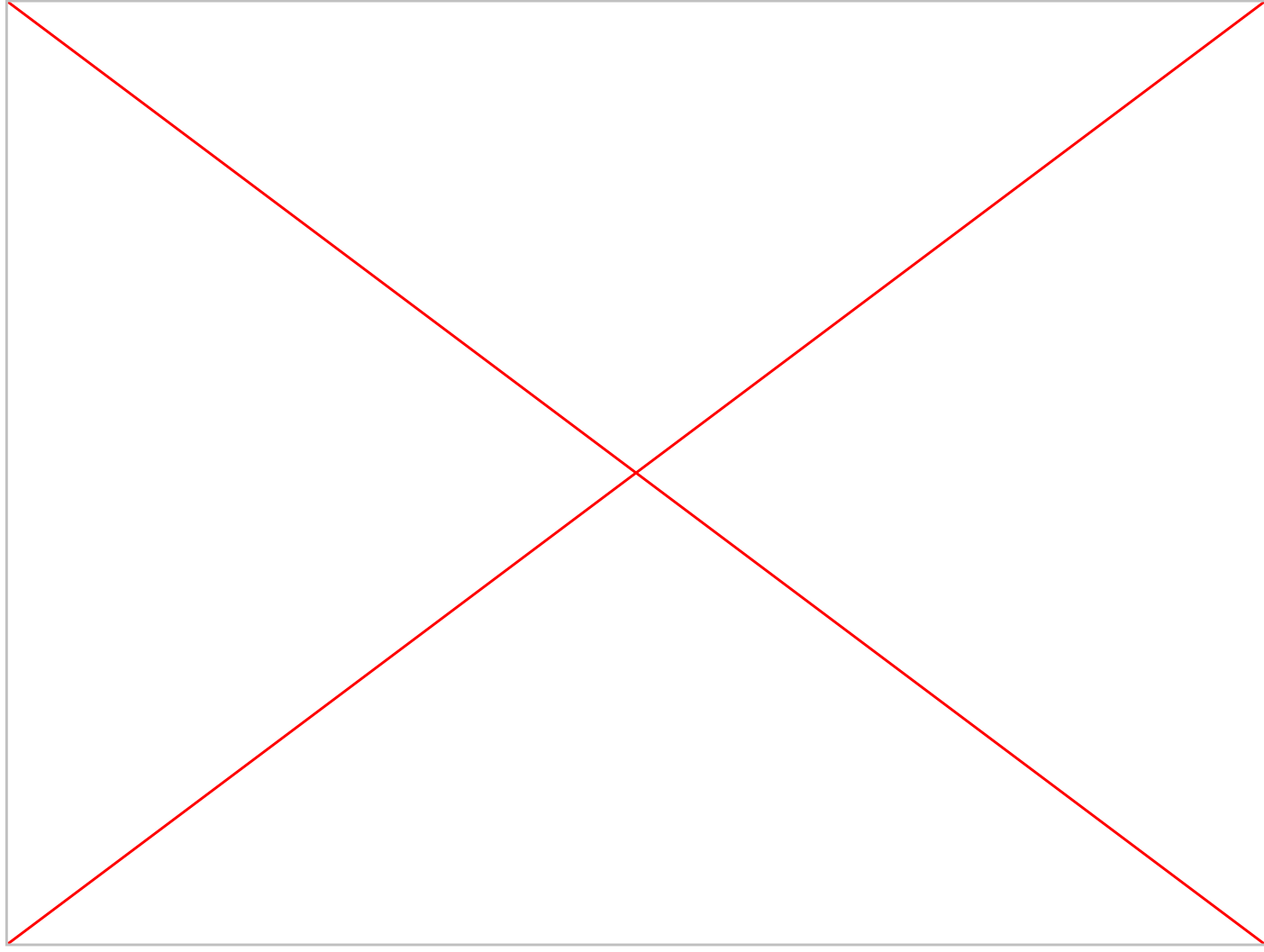
Symmetry below 70% → vibration motor + LED on injured brace



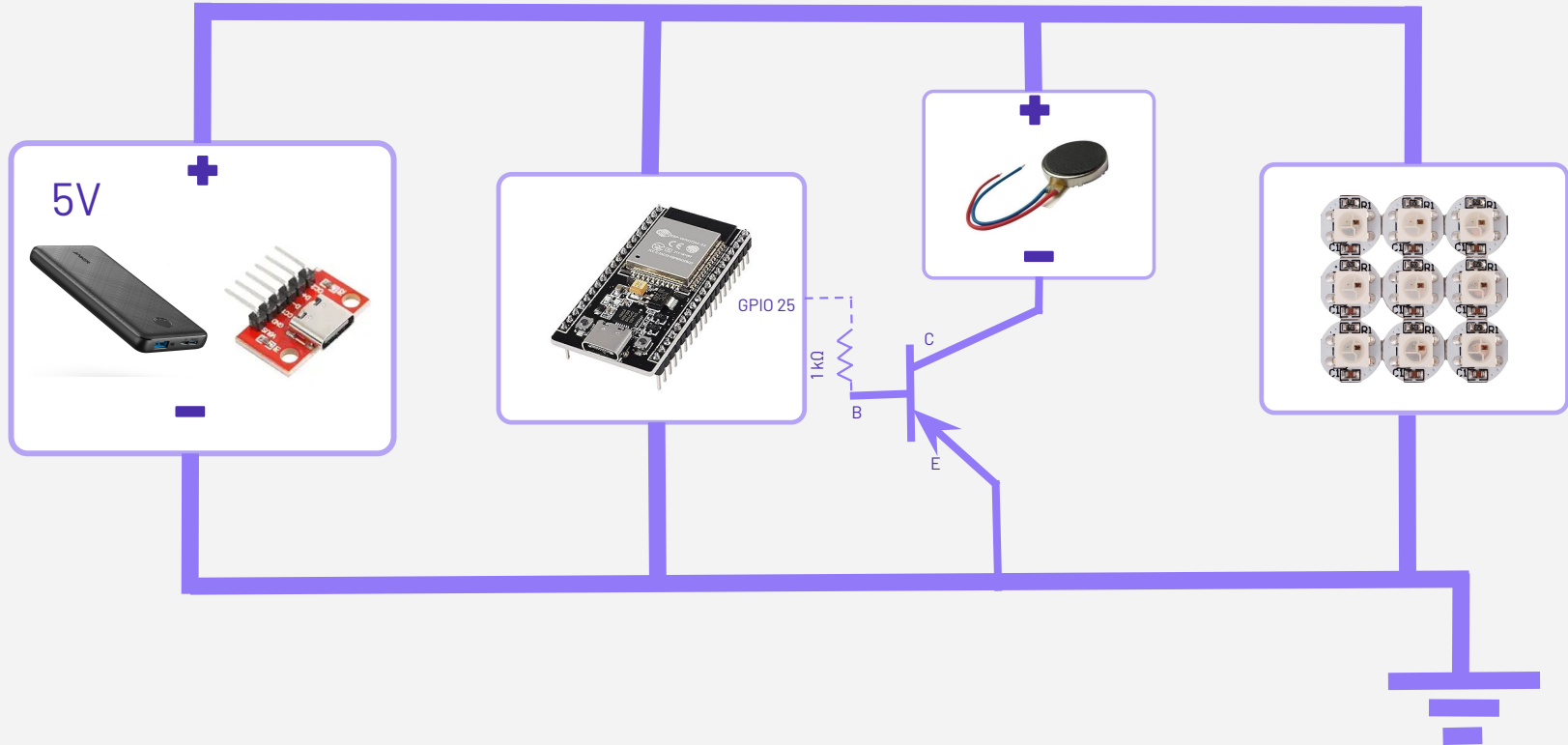
Live in browser

WebSocket pushes per-leg load + symmetry to the dashboard at 20 Hz

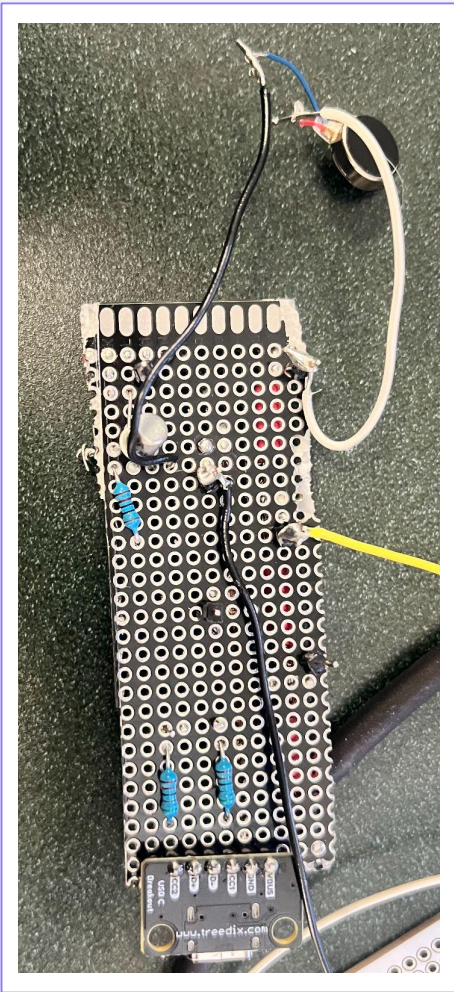
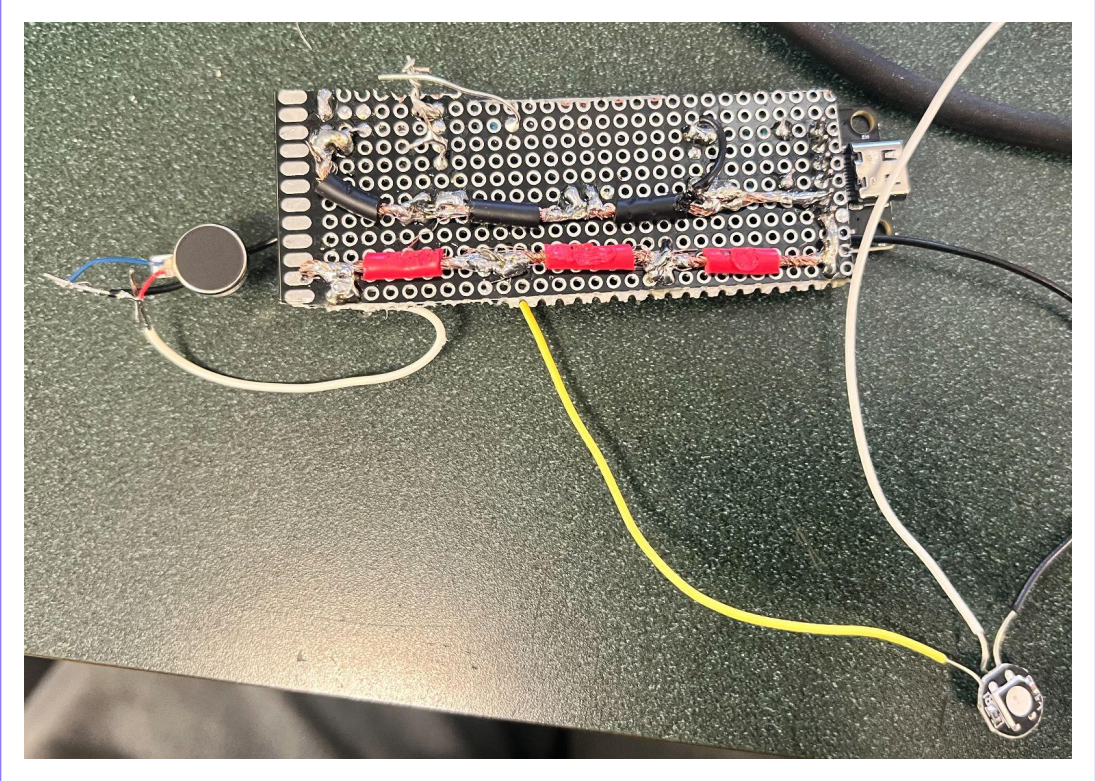
Weight Distribution



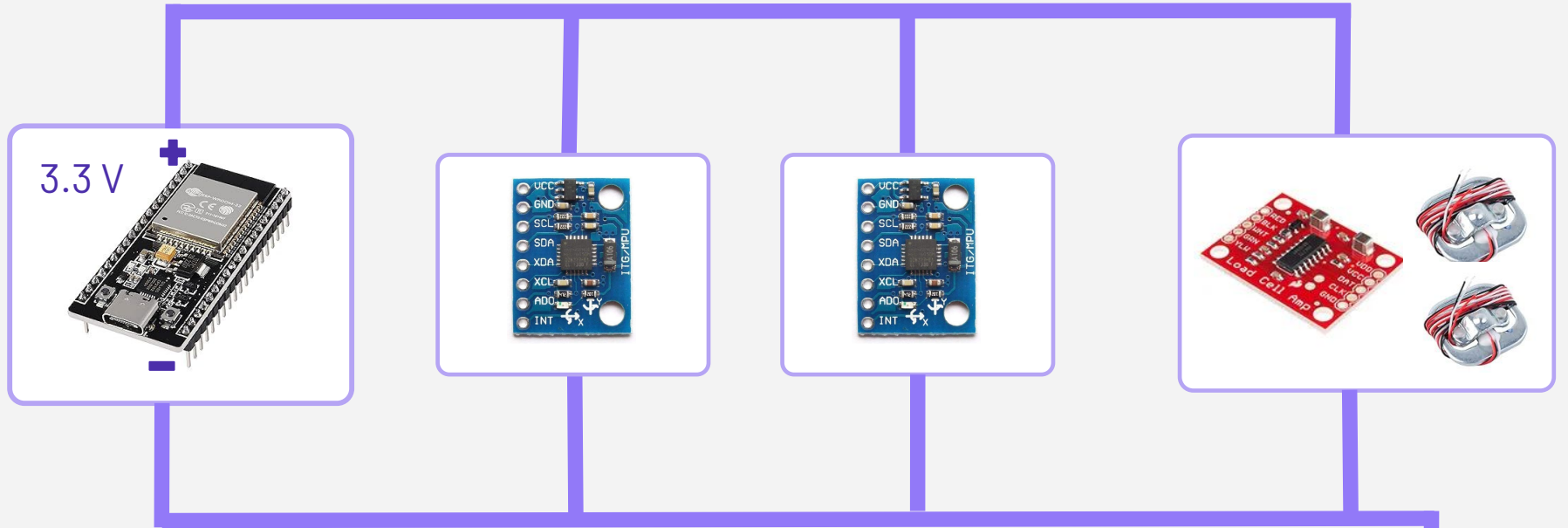
Power Distribution



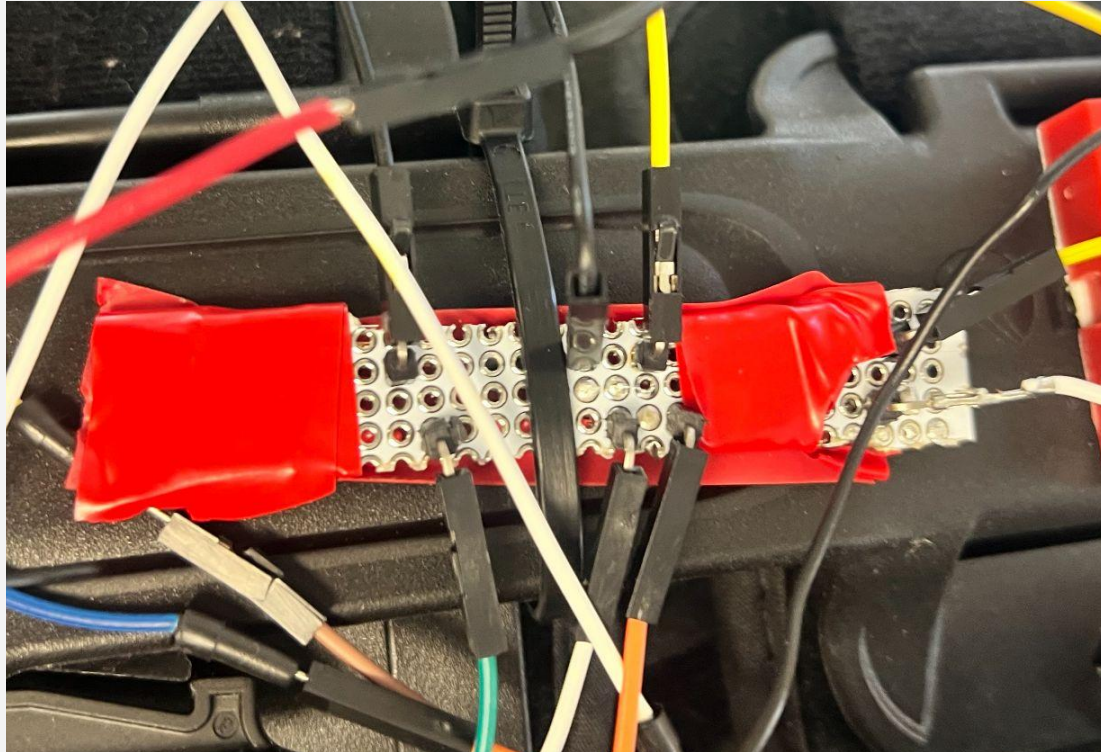
Power Distribution



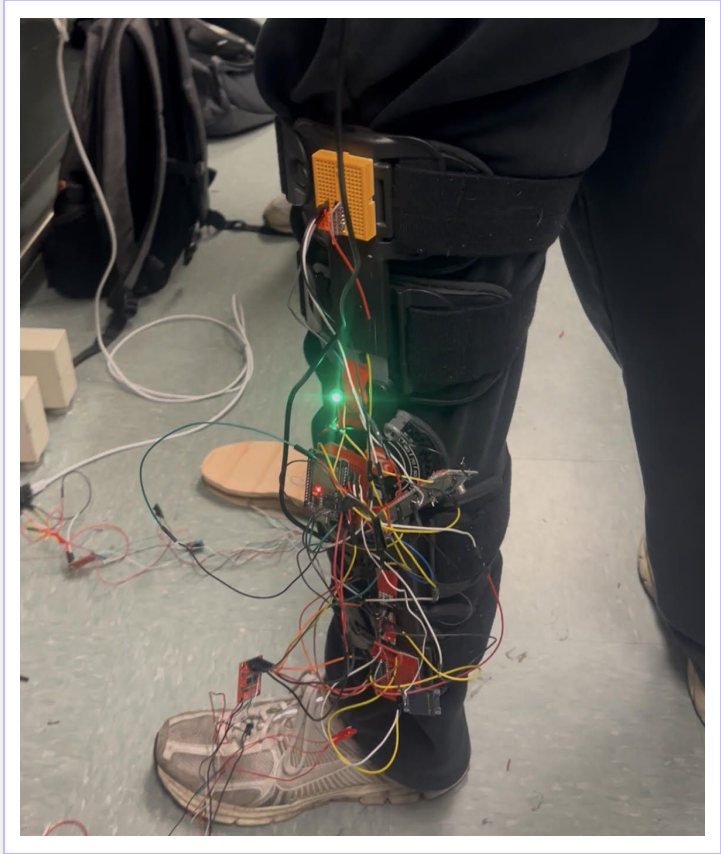
Power Distribution



Power Distribution



Our Final Design





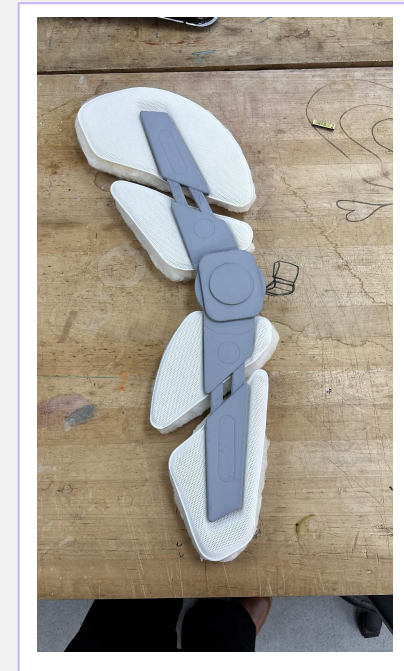
Brace Design — Render & Build

OTIS-rendered concept (left) and where the physical build stands today (right)



Concept render

by OTIS (USC industrial design team)



Physical build



Thank You

Questions?

