UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

Athletic Tracking Sensor

Team 38 (J.D. Armedilla, Ryan Horstman, Ethan Pizarro)

May 6, 2025

Agenda

1. Problem Statement
2. The Design
3. Final Design and Subsystems
4. Questions

Problem Statement

The Background

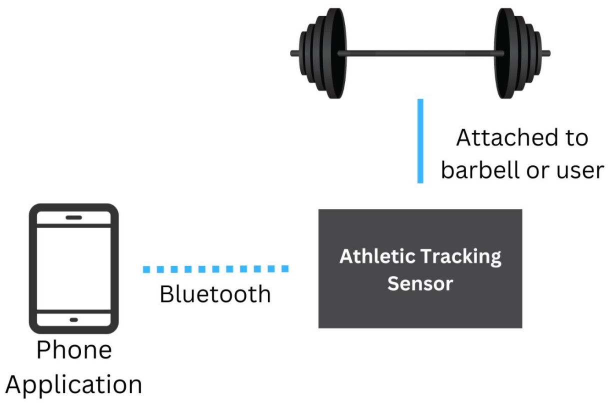
- Traditional weightlifting consists of varying heaviness of weight and number of repetitions
- Our project focuses on velocity-based training
  - The speed in which you lift weights

The Problem

- Current tracking sensors are expensive
- Current sensors convey information graphically only post-workout set
- Current sensors do not incorporate form tracking capabilities

Solution:

We created a cheap athletic tracking sensor that gives immediate feedback to the user mid-rep on both velocity goals and form safety via a vibration motor.

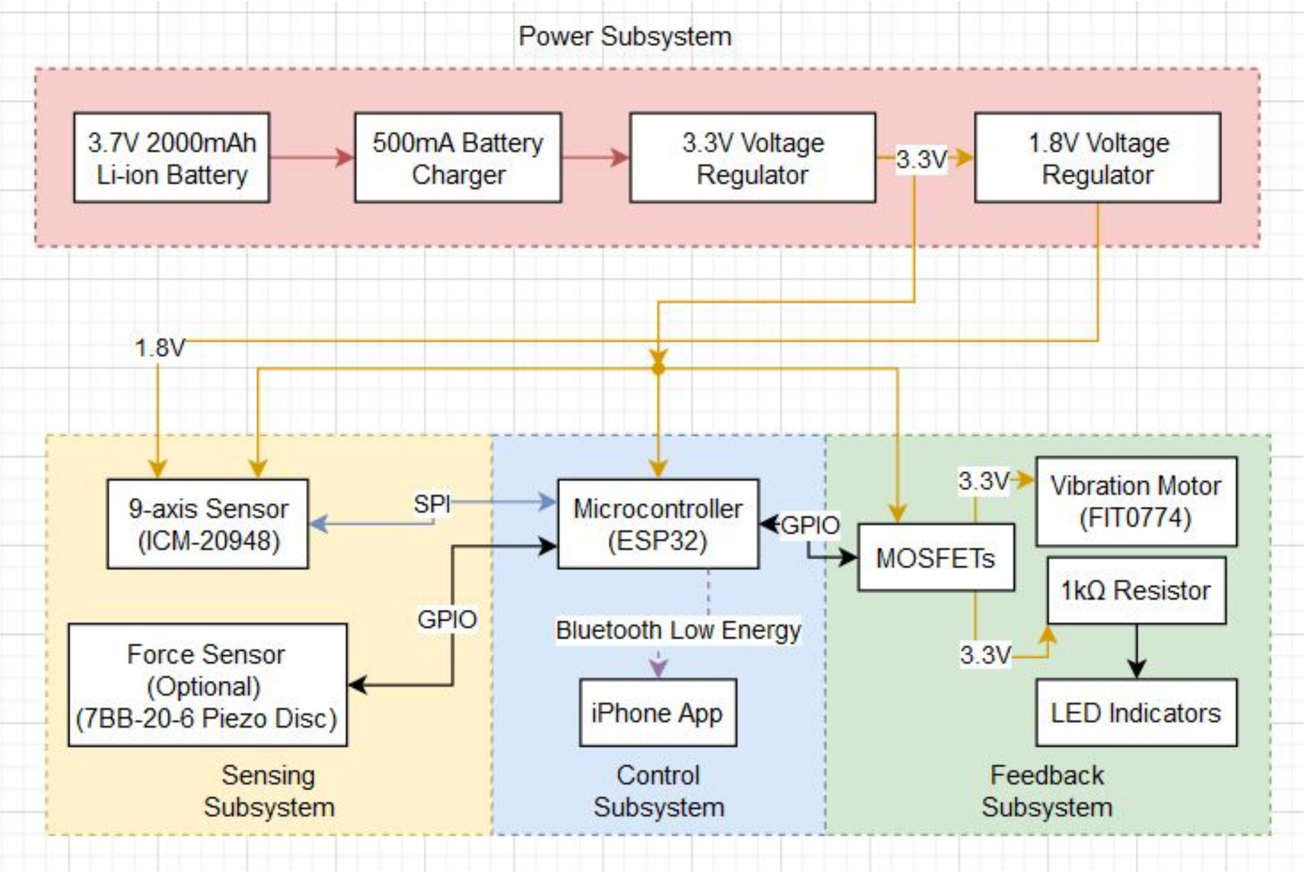


Measure Velocity, Detect Angle, and Collect Data

- Determine vertical velocity to hundredths of m/s
- Send/receive data via Bluetooth to our own iPhone app for easier processing
- Alert user immediately if velocity/angle thresholds are exceeded

The Design

Original Block Diagram



Original Block Diagram

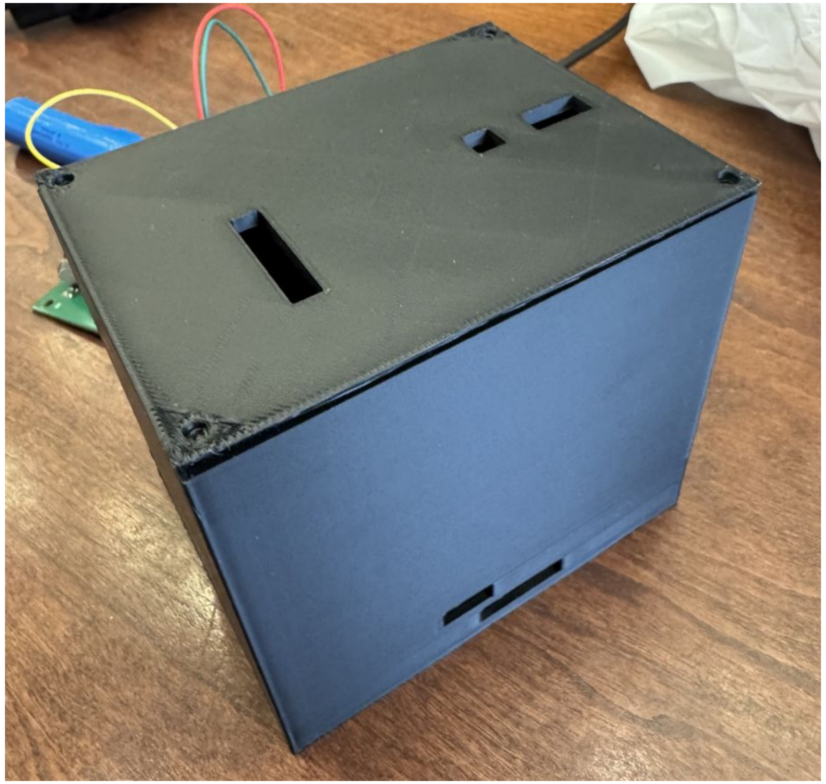
What Changed?

Sensing Subsystem

- Level shifter added (V_{DDIO} of ICM-20948 is ~1.8V while SPI pins of microcontroller output ~3.3V)

Feedback Subsystem

- MOSFET removed
- Button added to let user dictate when exercise recording starts/stops
- Switch connects/disconnects device from the battery
- 3D modeled basic PCB enclosure to allow outside access to switches, buttons, and LEDs



PCB enclosure

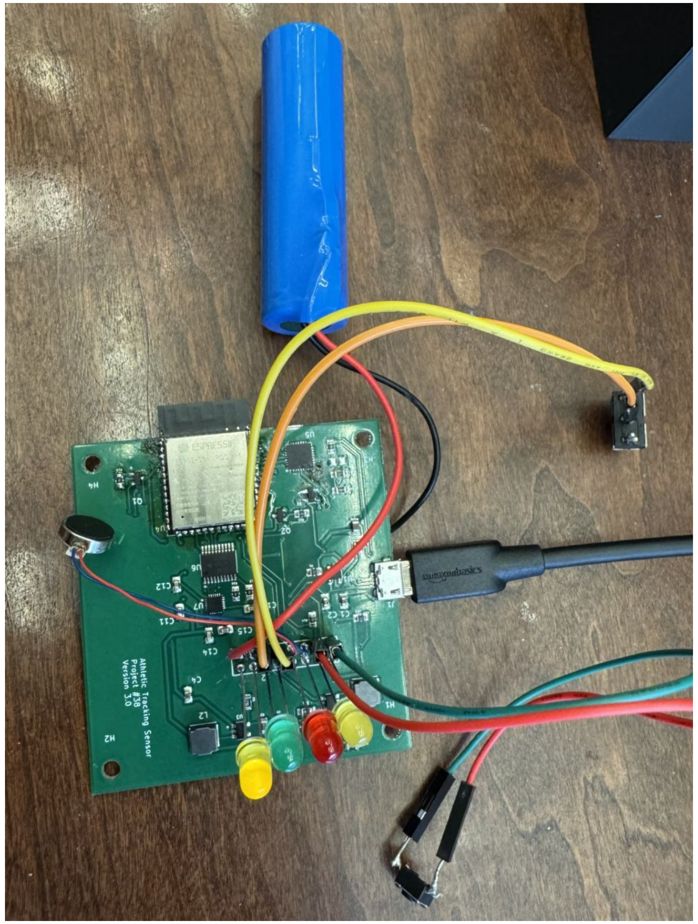
What Changed?

Power Subsystem

- Linear Li-ion charger for simpler soldering
- Buck converters for higher available current
- Micro USB port supports charging and data transfer at the same time

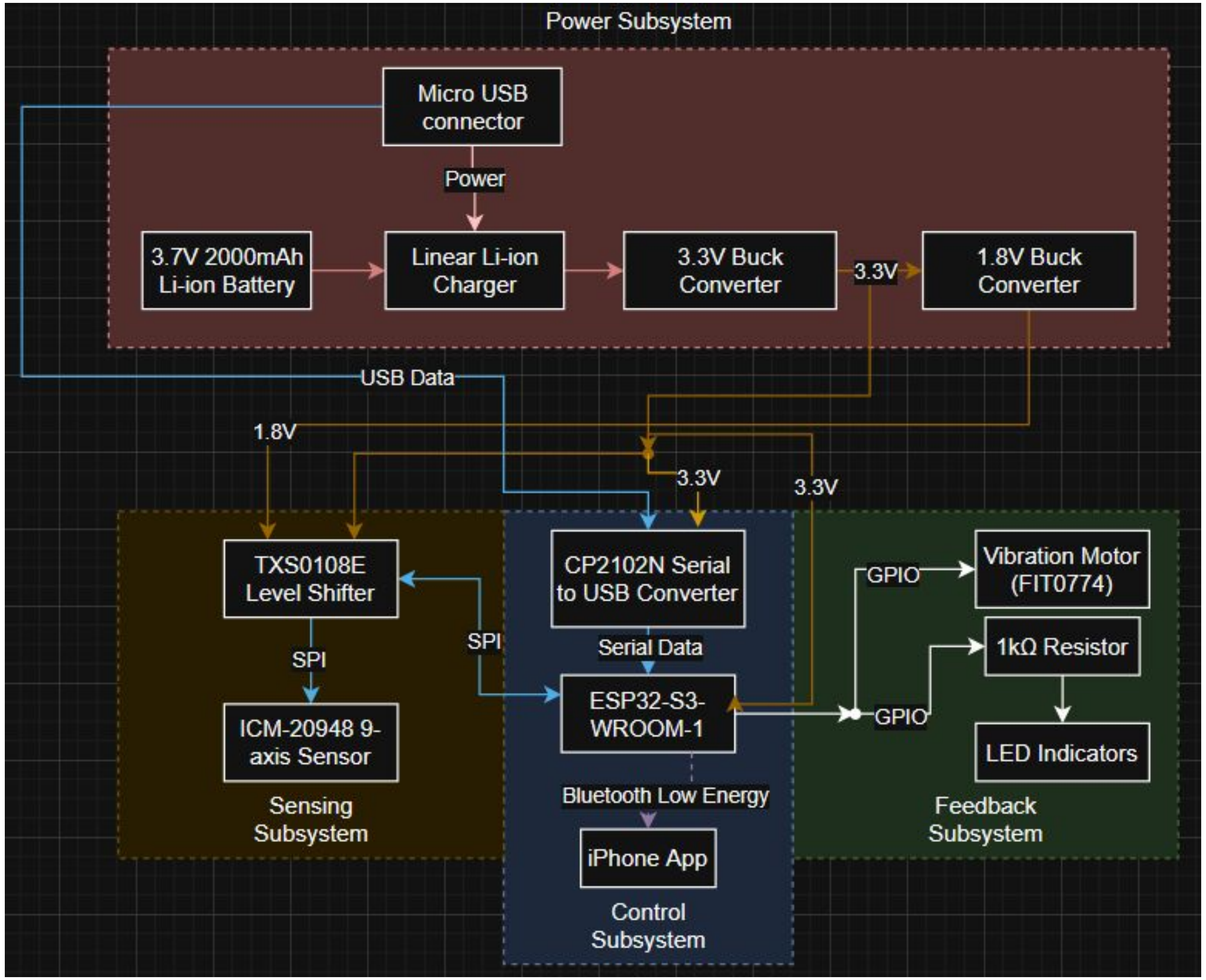
Control Subsystem

- CP2102N USB-to-Serial converter for programming
- ESP32-S3-WROOM-1 (S3 had pins too small to solder)
- iPhone app flow state changed to support Bluetooth functionality while enabling data display



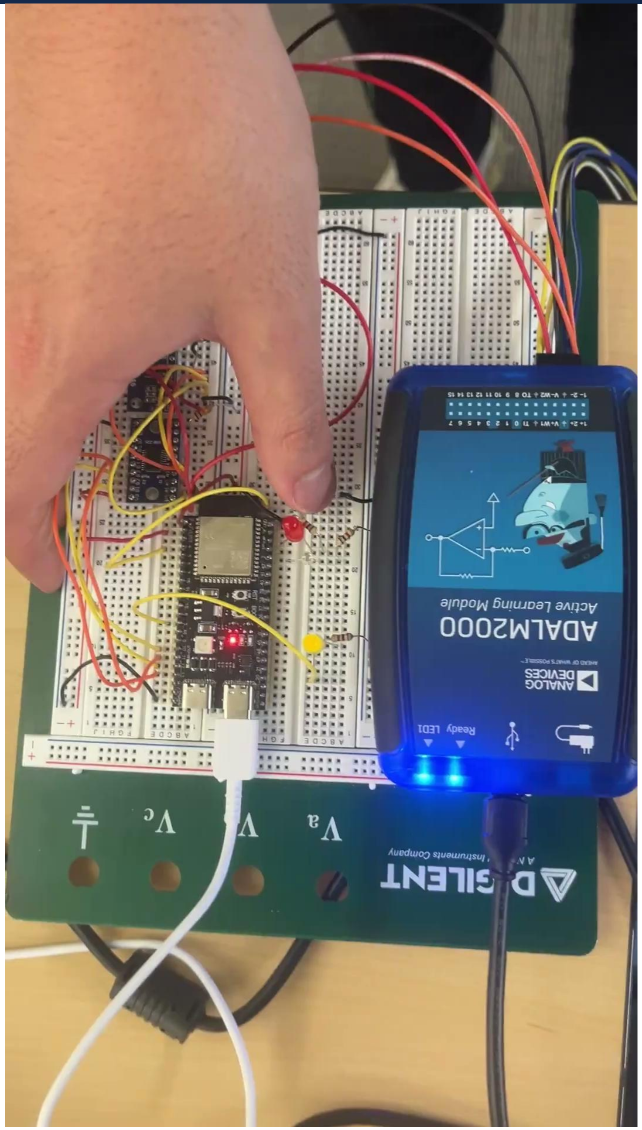
Final PCB

Final Block Diagram



Final Block Diagram

Demo Video

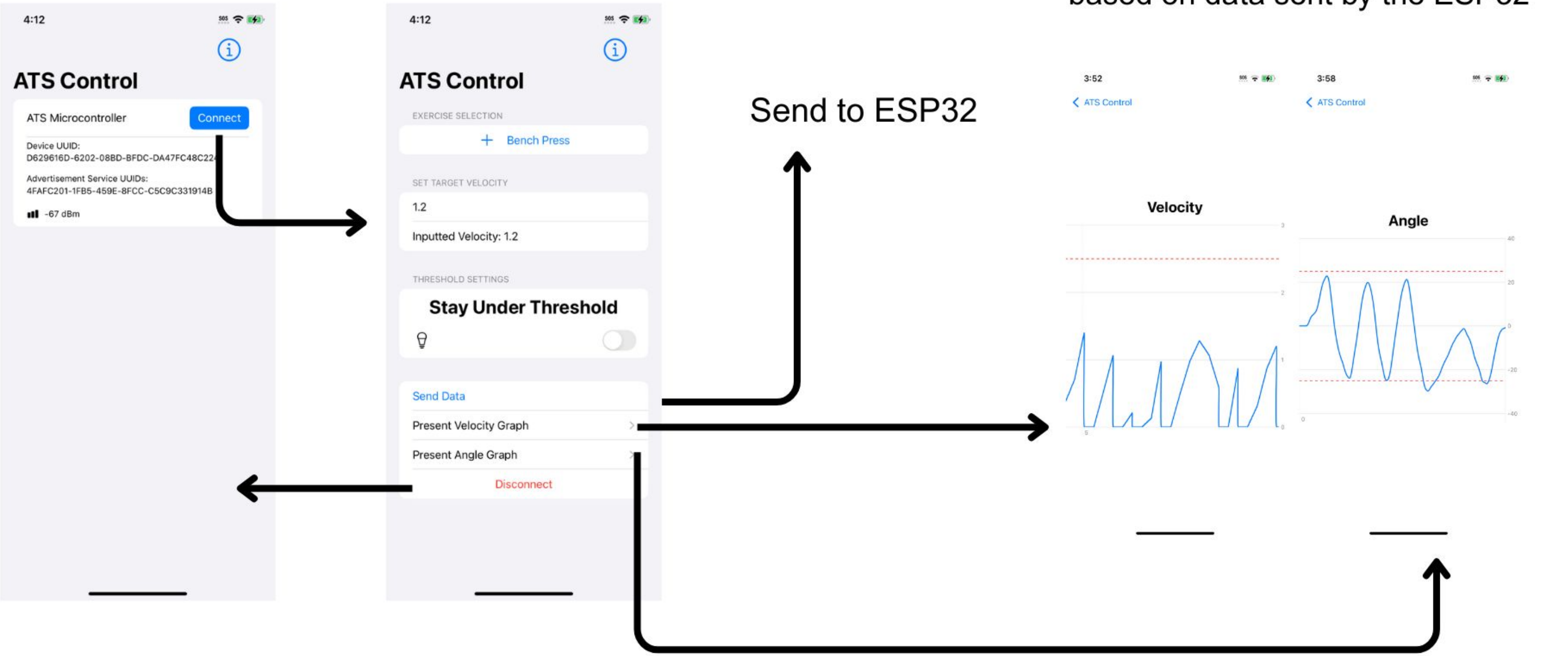


The Build: The App

Display available devices

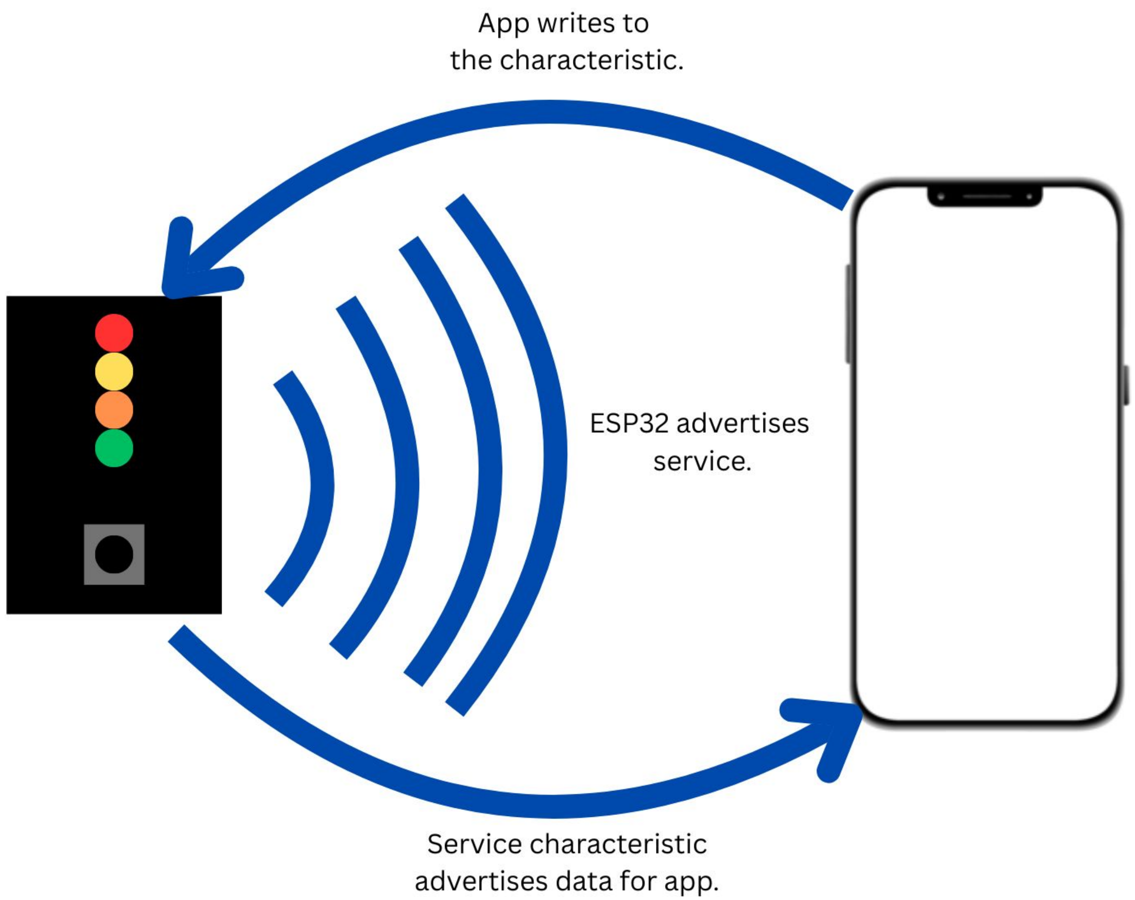
Control what information to send to ESP32

Velocity and angle/orientation graphs based on data sent by the ESP32



Connecting to ESP32

- ESP32 advertises a service
- Application searches for that service with the service UUID



Communicating Data

- Communication through a service's characteristic
  - Read and writes of the ESP32 are done through the characteristic

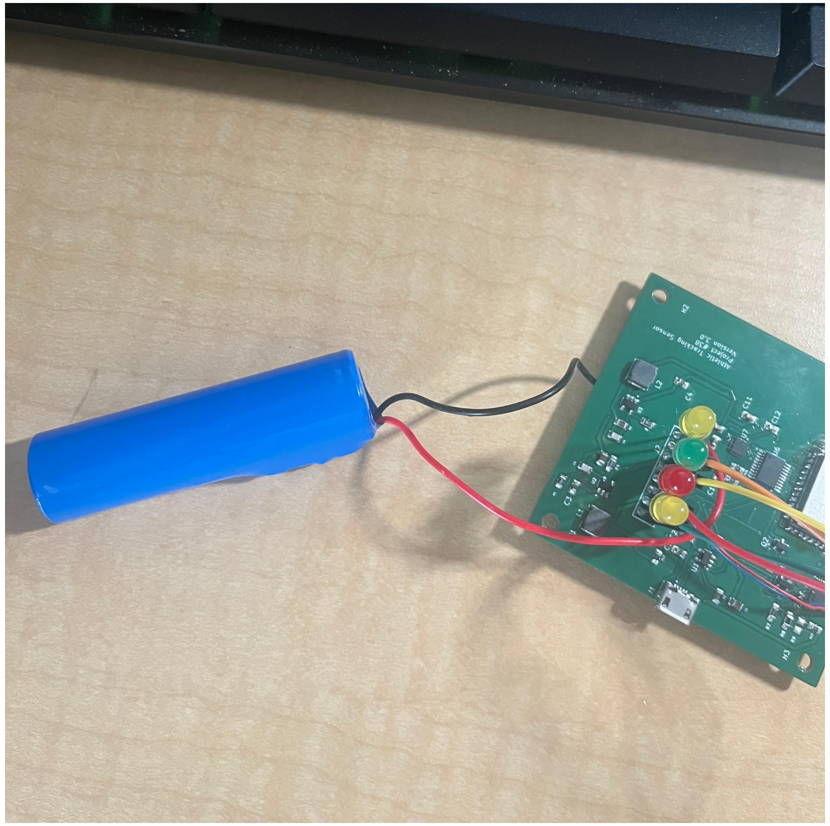
Power Supply and Delivery

Supply

- 3.7V 2000mAh Li-ion battery, linear Li-ion charger, and Micro USB port
- Charges within 6 hours with 2+ hour battery life

Delivery

- 3.3V buck converter (microcontroller, ICM-20948) and 1.8V buck converter (level shifter, ICM-20948 IO pins)



Battery connected to PCB

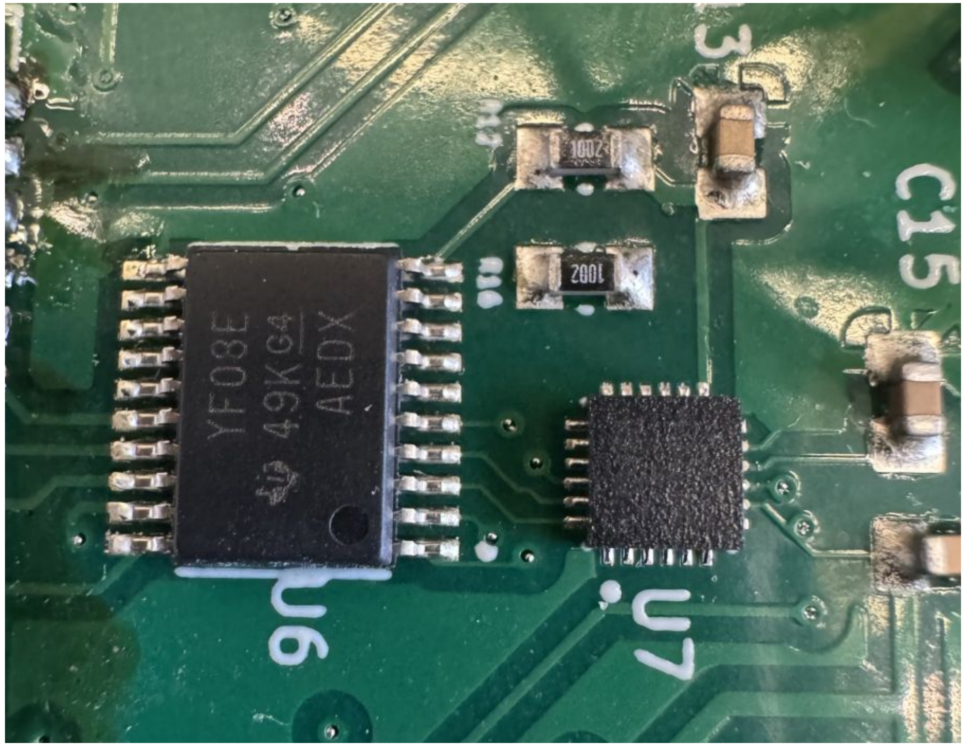
Final Design: Sensing Subsystem

9-Axis Sensor: ICM-20948

- Includes accelerometer, gyroscope, and magnetometer
- Communication via SPI protocol

Level Shifter

- SPI pins operate at 3.3V and ICM-20948 operates at 1.8V
- Level shifter makes SPI signals compatible with the IMU



Final Design: Data Analysis

Integration of Sensor Outputs

- Both Angle and Velocity values are calculated by integrating angular velocity and acceleration, respectively
- For acceleration, vertical acceleration due to gravity is picked up by the sensor, and had to be accounted for
- The magnitude of total change in velocity change is used instead of a particular direction for output velocity

Deadband Filtering

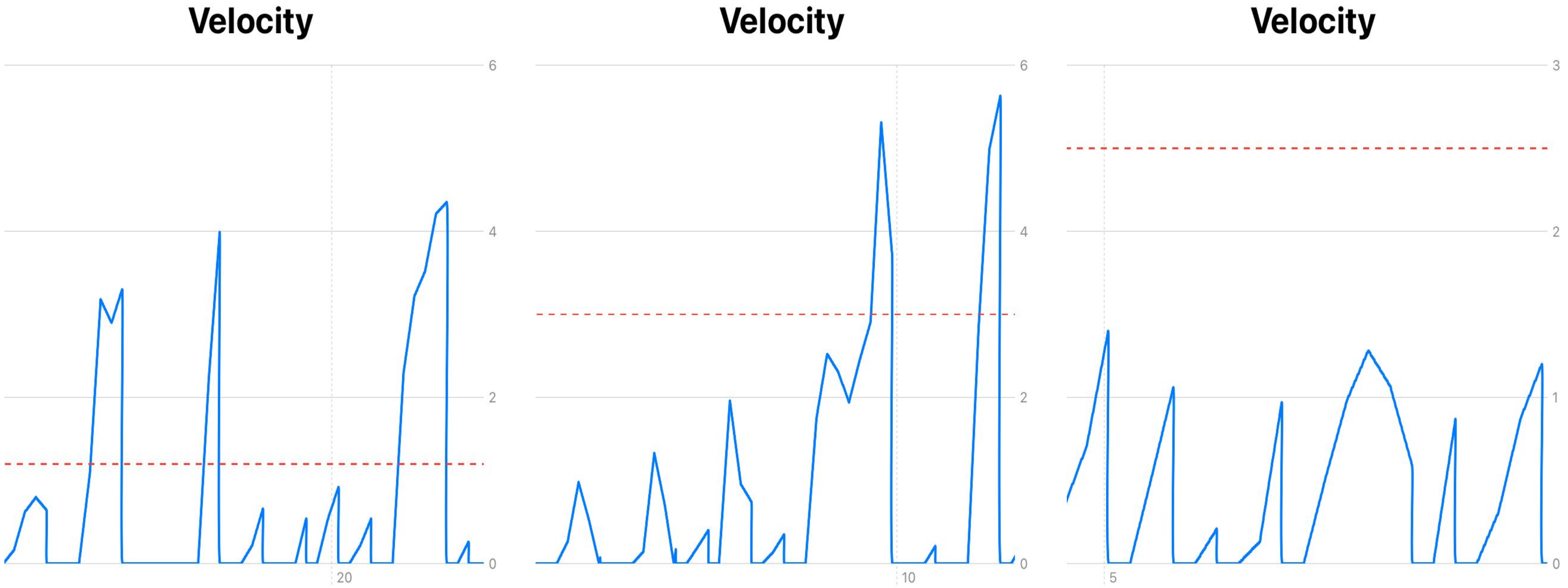
- Both accelerometer and gyroscope are sensitive to vibration noise that produces drift in angle and velocity outputs

```
myIMU.getGyrValues(&gyr);

float currentTime = millis();
float deltaTime = (currentTime - gyr_lastTime) / 1000.0;
gyr_lastTime = currentTime;

// Apply deadband to gyroscope raw data
float gyrDeadband = 2;
if (abs(gyr.x) < gyrDeadband) gyr.x = 0;
if (abs(gyr.y) < gyrDeadband) gyr.y = 0;
if (abs(gyr.z) < gyrDeadband) gyr.z = 0;

// Integrate gyro readings to get angles
angleX += gyr.x * deltaTime;
angleY += gyr.y * deltaTime;
angleZ += gyr.z * deltaTime;
```

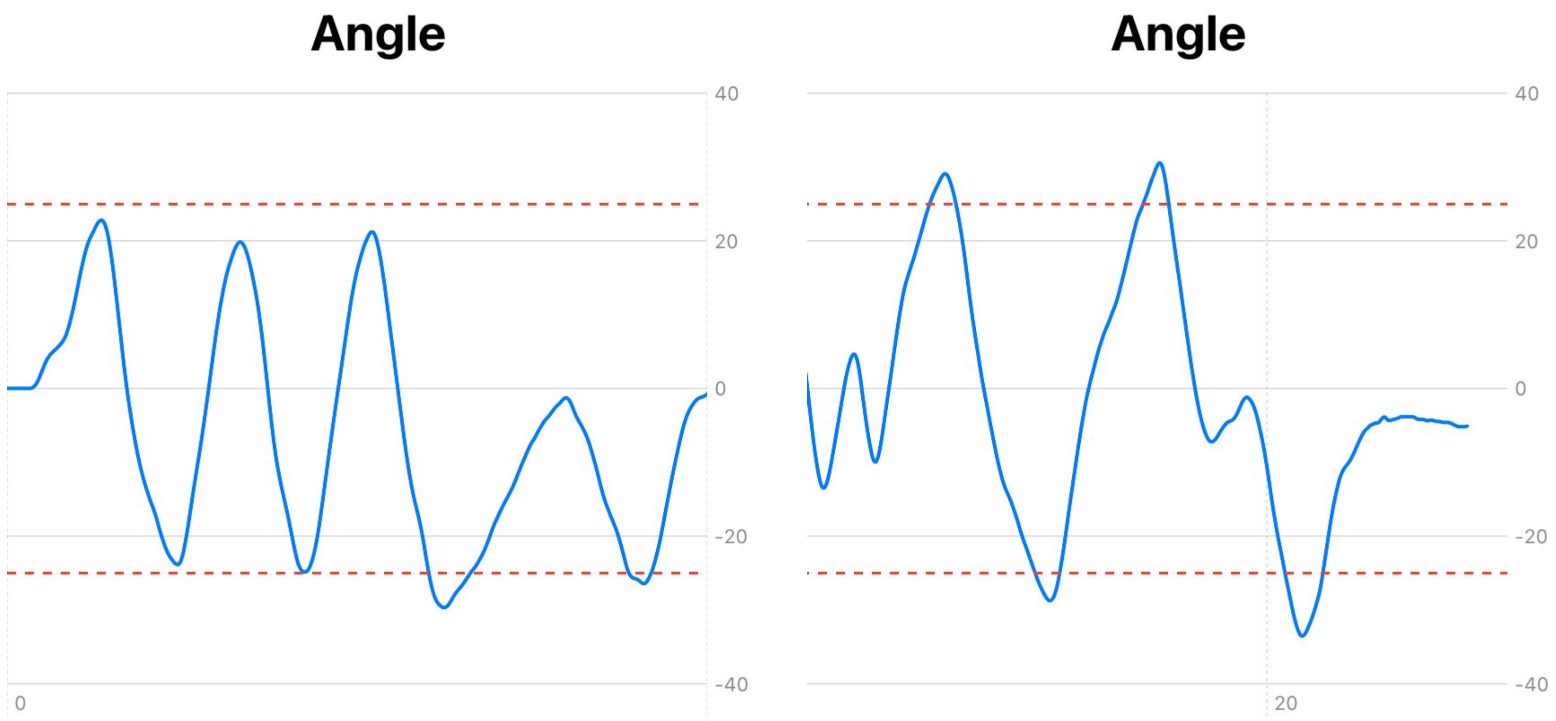
$$v = v_0 + a\,t$$

$$|v| = \sqrt{x^2 + y^2 + z^2}$$

Results: Velocity Tracking



Results: Form Tracking



Final Design: Control Subsystem

Sensor Processing and Programming

- Utilizes native SPI and Bluetooth Low Energy functionality to work with ICM-20948 and iPhone app
- Determines when GPIO pins activate, driving feedback subsystem
- Listens for “Record Data” button input to start monitoring velocity/angle thresholds

Figure 11: USB-to-Serial converter (U5) and microcontroller

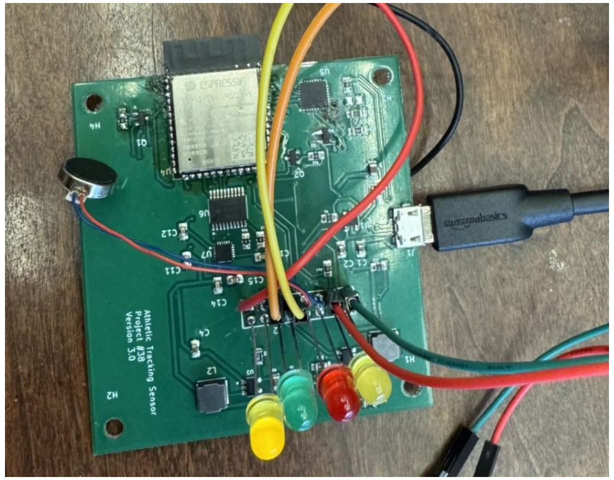
Microcontroller Tasks

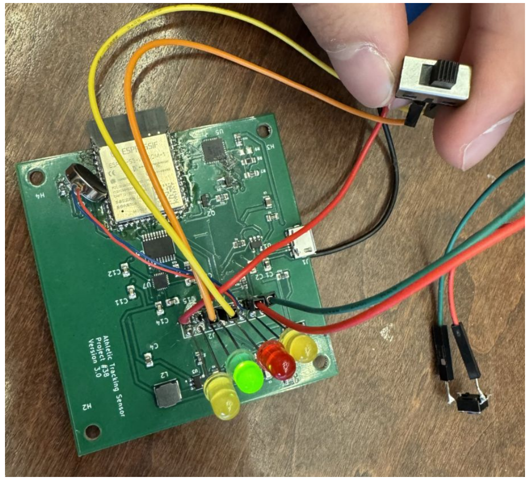
- Read/write to ICM-20948
- Activate LEDs and vibration motor
- Send/receive data to/from iPhone app

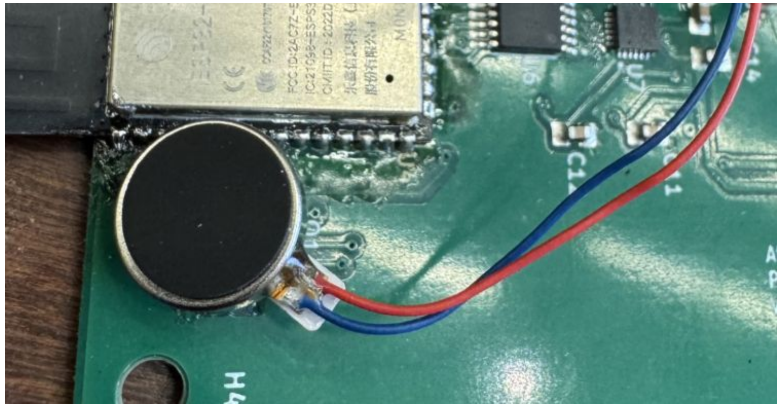
Final Design: Feedback Subsystem

Haptic Feedback

- Vibration Motor actuating with user notification LED







Indication LEDs

- Power-ON
- Device Charging
- State of Device (waiting or taking data)
- User Notification

Issues with Programming the ESP32-S3-WROOM-1

- V_{REGIN} and V_{DD} left unconnected in the PCB's schematic on the USB-to-Serial converter
  - Misunderstood from datasheet that V_{BUS} would provide its power
- Solution: connect 3.3V power to V_{REGIN} and V_{DD}

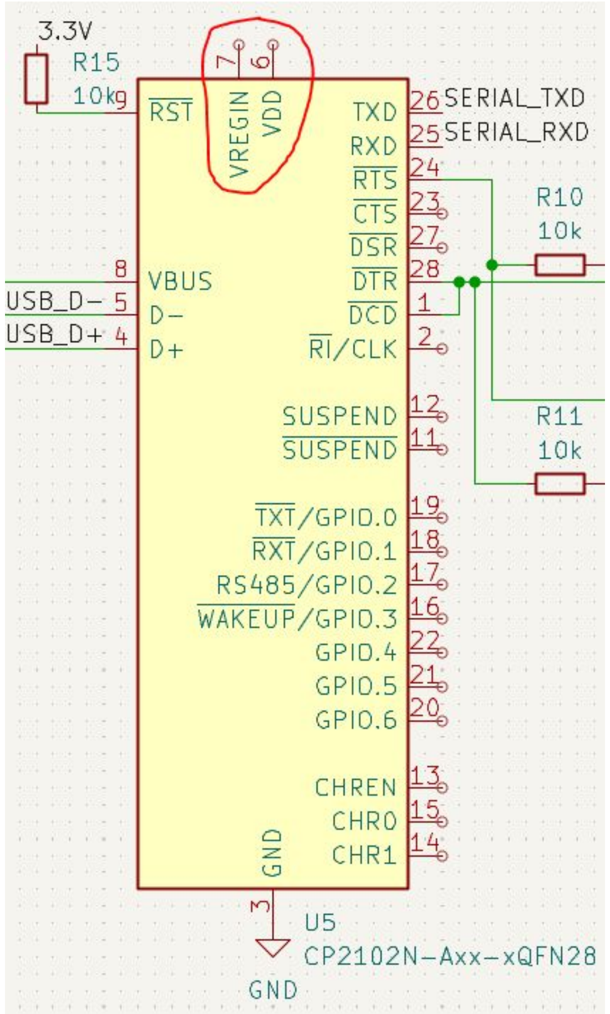


Figure 16: V_{REGIN} and V_{DD} pins

Protecting Users and Their Data

- Bluetooth Low Energy encryption
- Preventing Injuries While Testing
- Ensuring User Movement
- Dissipating Heat
- Makes Velocity-Based Training More Accessible

Conclusion

Accomplishments & Future Work

Summary of Accomplishments

- Created a working rechargeable battery power supply for a wearable device
- Calculated velocity and angles from a 9-axis sensor's raw data
- Provided live feedback during exercises, beating out industry standards
- Developed an app that utilizes bluetooth communication to display data

Future Work

- Get PCB fully functional
- Optimize space usage (decreasing enclosure size)

Questions

References

[1] T. T. C.S.C.S, “What You Need to Know About Spotting in the Gym,” Men’s Health, Jul. 20, 2020. https://www.menshealth.com/fitness/a33366411/how-to-spot-weight-lifting/

[2] “Barbell Bench Press Overview,” fitnessprogramer.com. https://fitnessprogramer.com/exercise/bench-press/ (accessed May 1, 2025).

[3] “Back Squat,” dmoose.com. https://www.dmoose.com/blogs/quads/back-squat (accessed May 1, 2025)

[4] IEEE, “IEEE Code of Ethics,” ieee.org. https://www.ieee.org/about/corporate/governance/p7-8.html (accessed May 4, 2025).

The Grainger College of Engineering

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN