Precision Dumbbell Assistant

ECE 445 Design Document - Spring 2024

Project #40

Cole Trautman, Ronit Kumar, Ellie Beck

Professor: Arne Fliflet

TA: Douglas Yu

1 Introduction

1.1 Problem

Many gym goers struggle to maintain proper form during their workouts with dumbbells, which is why they rely heavily on exercise machines. Maintaining proper form is important for two reasons. Bad form can increase the risk of injuries, especially with heavier weight, and can reduce the efficiency of the exercise, making it less effective at building muscle and strength [1]. Many people want to have some sort of at-home gym so that they can work out in the comfort of their own home and maybe avoid paying a gym membership fee, but they will miss out on all the equipment that a full gym has to offer. If you are trying to construct an at-home gym, often all you will have, at least to start, is a set of dumbbells and a bench. Hence, there should be a relatively cost-effective way to help people maintain proper form even when they just use dumbbells so that they can get the maximum benefit from their exercise.

1.2 Solution

We will design a device that will track the user's arms to ensure that their form is correct. Our design will use 3 6-axis (accelerometer and gyroscope) IMU (inertial measurement unit) sensors on each arm to calculate the position of each arm. There will be two small sensor boards located on the lower arm and shoulder, and a larger main board with another sensor and the main processor on the upper arm. This will allow us to track the movement of these three parts of the arm relative to each other and determine whether the movement is correct or not. We will develop an algorithm to detect the correct movement for an exercise, as determined by experts in the field. In order to keep the scope of this project reasonable, we will just develop an algorithm for bicep curls, with the ability to expand the other exercises if desired. If incorrect form is detected, the user will be notified with a buzzer, and more detailed information will be provided through a smartphone app. The app will be an Android app that connects to the processor on each arm via Bluetooth, and will allow the user to view their past set and see the number of reps, speed, and areas in which incorrect form was detected.

1.3 Visual Aid

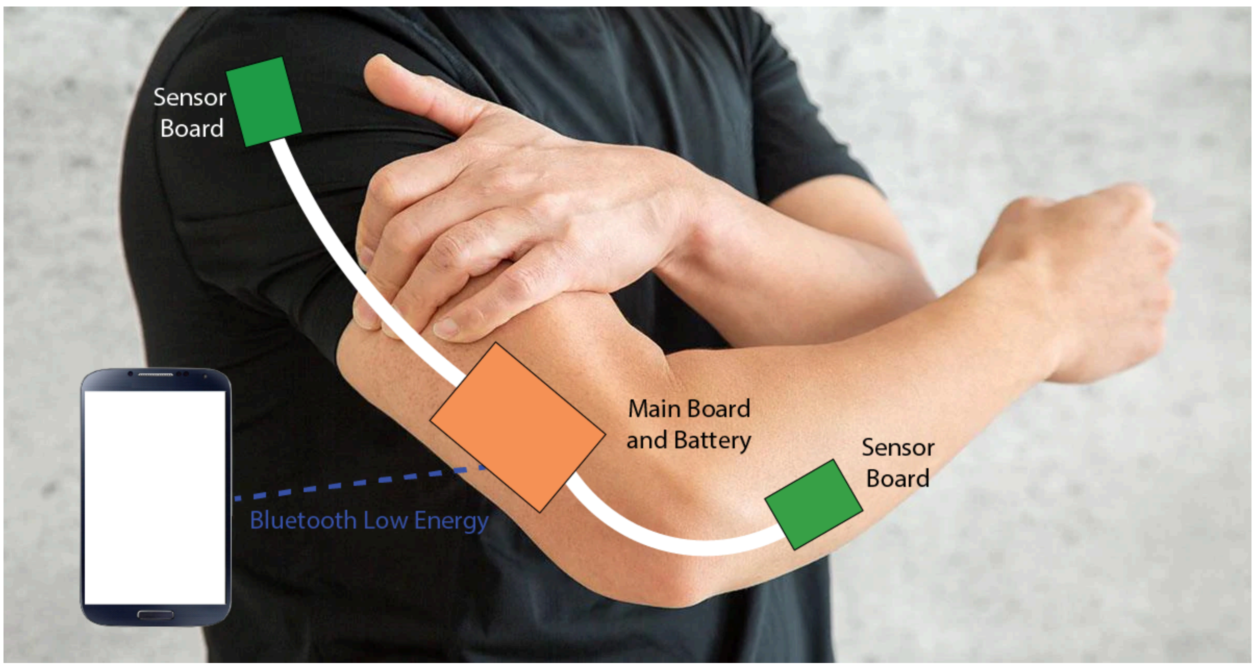


Figure 1: Visual aid of device positioning

Our device will have sensors attached to the user's arm in three locations: one on the outside shoulder, one on the back of the tricep, and one on the bottom of the forearm. If worn correctly, the cable going from the tricep to the forearm will go directly over the elbow. The board on the back of the tricep will also have the processor and the battery, so it will be a little bit bigger and heavier. All three boards will be attached to the user using elastic straps, and foam tape will be used on the back of the boards to help them stay in place.

1.4 High Level Requirements

In order for our project to be successful, we must meet the following three requirements:

- Our device needs to be accurate and consistent in motion and form analysis. Each sensor should be able to correctly calculate its position and orientation relative to a starting point within a tolerance of ±5% of the true values. The result should also be repeatable, meaning that if we perform the same motion multiple times, it should be either categorized as correct or incorrect every time. We should be able to move our arms at the same distance and angle that we determine from our research of an online fitness expert and the feedback should be positive.
- The device must be able to give the user feedback quick enough and loud enough to diagnose incorrect form. The entire system should read sensor data, analyze it, and provide feedback in no more than 50 ms, meaning that feedback should be provided at a minimum of 20 Hz.
- Our device should not restrict the movement of the user in any significant way. Its weight must be negligible (< 200g). All the connections must be flexible enough and of appropriate length so that the device fits most arms without getting in the way.

2 Design

2.1 Block Diagram

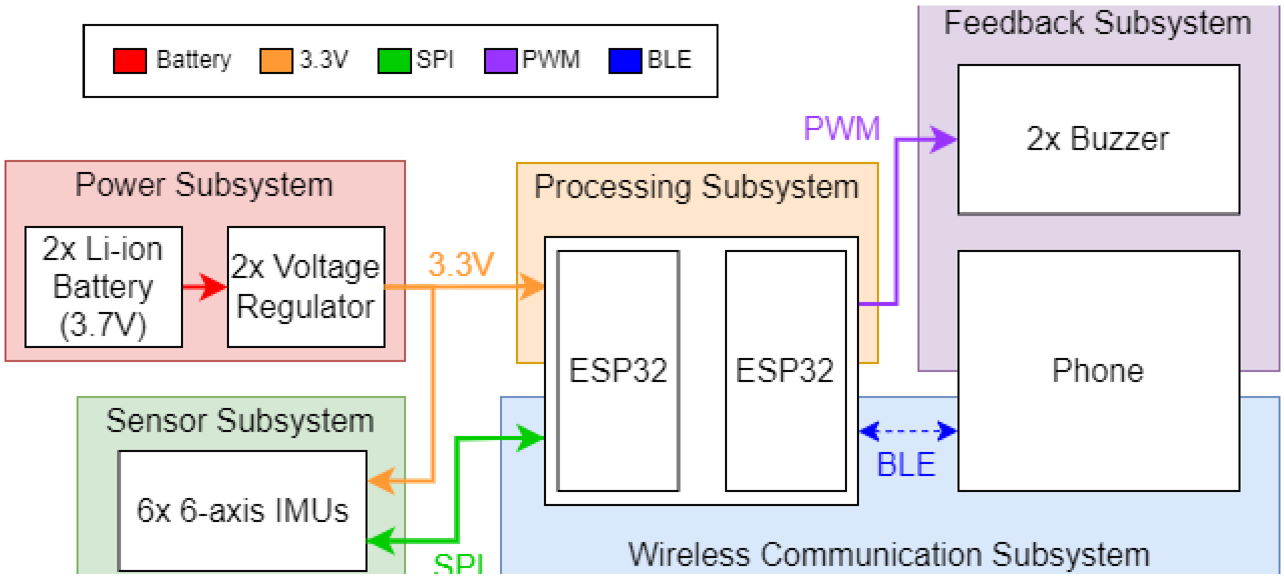


Figure 2: Precision Dumbbell Assistant Block Diagram

Our device has 5 subsystems: Sensing, Processing, Wireless Communication, Feedback, and Power. The sensing subsystem is responsible for providing all the data necessary to calculate the position of the user's arm, namely acceleration and angular velocity data. This data is fed into the processing subsystem, which is responsible for converting acceleration and angular velocity into position and orientation. The processing subsystem is also responsible for calculating whether the position and orientation of the sensors are within the acceptable bounds for the exercise. The wireless communication subsystem is responsible for establishing and maintaining a wireless link between the device and a smartphone that is used to send data to a smartphone app. The feedback subsystem is responsible for providing feedback to the user, both through a buzzer on the device and through the smartphone app. Finally, the power subsystem is responsible for providing the other subsystems with the correct voltage.

2.2 Physical Design

The main physical component of the design will be the battery enclosure. The battery we have chosen is pretty light (22g) [2], so we think that mounting it on the user's arm is feasible. The enclosure will be 3D printed out of PLA, and will sit underneath the main board, in between the user's arm and the main board. The main board will be attached to the enclosure with 4 M3 screws, and the battery will friction fit into the enclosure. Each board will be attached to the user's arm using an elastic strap, which will be connected to cutouts on the PCBs. Additionally, each board will have foam tape on the bottom, which will protect the user from any THT components sticking through the board, as well as provide additional grip to keep the boards in the correct position.

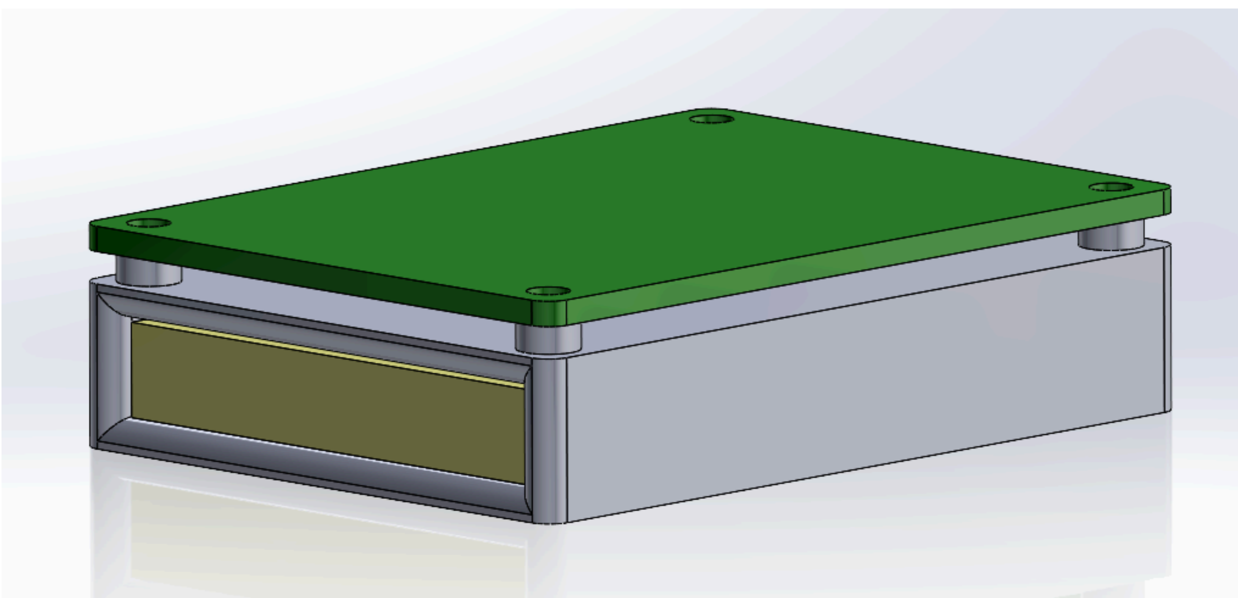


Figure 3: Assembly of battery, enclosure, and main PCB

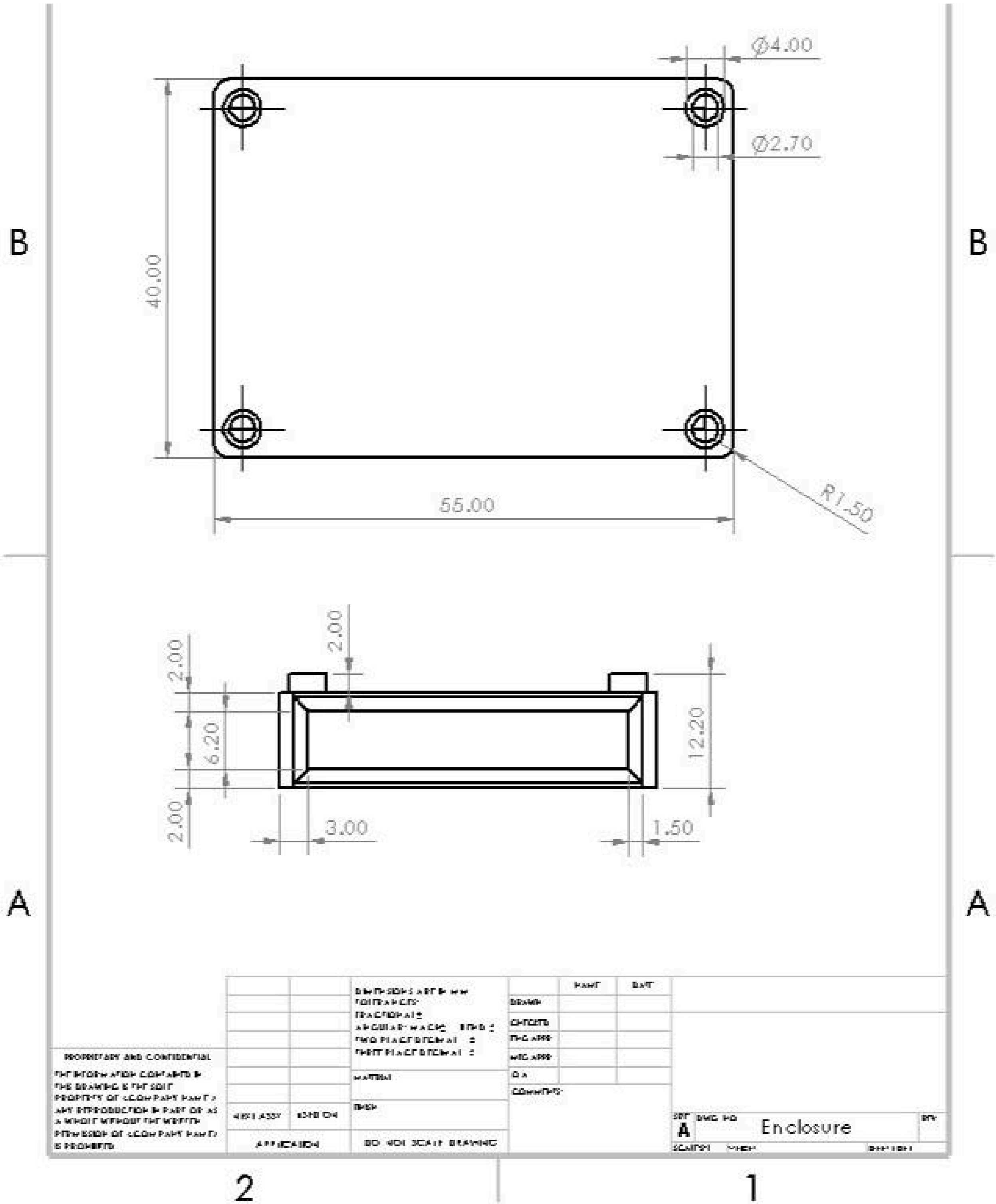


Figure 4: Enclosure drawing (units in mm)

2.3 Subsystems

2.3.1 Sensing Subsystem

This subsystem is responsible for collecting the raw data needed to track the user's arm. It contains the LSM6DSMTR IMU sensors (accelerometer and gyroscope) we will use to track each part of the arm [3]. As mentioned before, the sensors will be located on the lower and upper arm, as well as the shoulder, which should allow us to accurately track the entire arm and dumbbell. The sensors will send raw data over SPI to the processing subsystem and receive 3.3V from the power subsystem. There are only two available SPI peripherals on the ESP32 microcontroller we have selected, so we have decided to poll each sensor individually using one SPI and multiple chip-selects. This is not as optimal as having a dedicated set of pins for each sensor, but it should still allow us to hit our required performance. This subsystem also includes the physical wiring from the main board to the separate sensor boards. Each connection will need at least 6 wires: 3.3V, GND, MISO, MOSI, CS, and SCLK. We might also use an interrupt pin on each IMU, which would bring the total to 7. This block is important for all three of our high-level requirements. More details on the specific requirements and verifications can be found in the table below.

Requirement & Verification Tables:

Requirement
Each IMU sensor should provide acceleration and angular velocity with an accuracy of ±5%
Verification
Equipment

LSM6DSMTR IMU sensor, ESP32 development board, flat surface rotates at a known angular velocity

Test Procedure

Accelerometer:

1. Connect IMU sensor to ESP32 development board using manual.
2. Place the IMU sensor on a hard flat surface with the chip facing up.
3. Request accelerometer readings using the ESP32 microcontroller.
4. Record the provided accelerometer data.
5. Repeat steps 3 and 4 5 times to gather enough data.
6. Repeat steps 2 through 5 for each of the 5 remaining axes (see datasheet pg. 20 for acceleration axes relative to package)
7. Each value should be within ±5% of the desired value. The desired value should be 1.0 with the sign of the axis pointing straight down. For example, the correct value for testing in the +Z direction (chip facing up) is X = 0.0g, Y = 0.0g, Z = -1.0g.
8. If all readings are within ±5% of the desired value, the test is successful.

Gyroscope:

1. Connect IMU sensor to ESP32 development board using manual.
2. Place the IMU sensor on the spinning surface with the chip facing up, and secure it so that it stays in place.
3. Start spinning the surface at a known angular velocity, and then request gyroscope readings using the ESP32 microcontroller.
4. Record the provided gyroscope data.
5. Repeat steps 3 and 4 5 times to gather enough data.
6. Repeat steps 2 through 5 for each of the 5 remaining axes (see datasheet pg. 20 for gyroscope axes relative to package)
9. Each value should be within ±5% of the desired value. The desired value should be the angular velocity of the spinning surface, with counter clockwise being positive. For example, testing with the chip facing up and spinning counterclockwise should result in a positive angular velocity that is close to the angular velocity of the spinning surface.
10. If all readings are within ±5% of the desired value, the test is successful.

Presentation of Results

Record all readings in separate data tables in the notebook for the accelerometer and gyroscope, and take note whether each test passes or not.

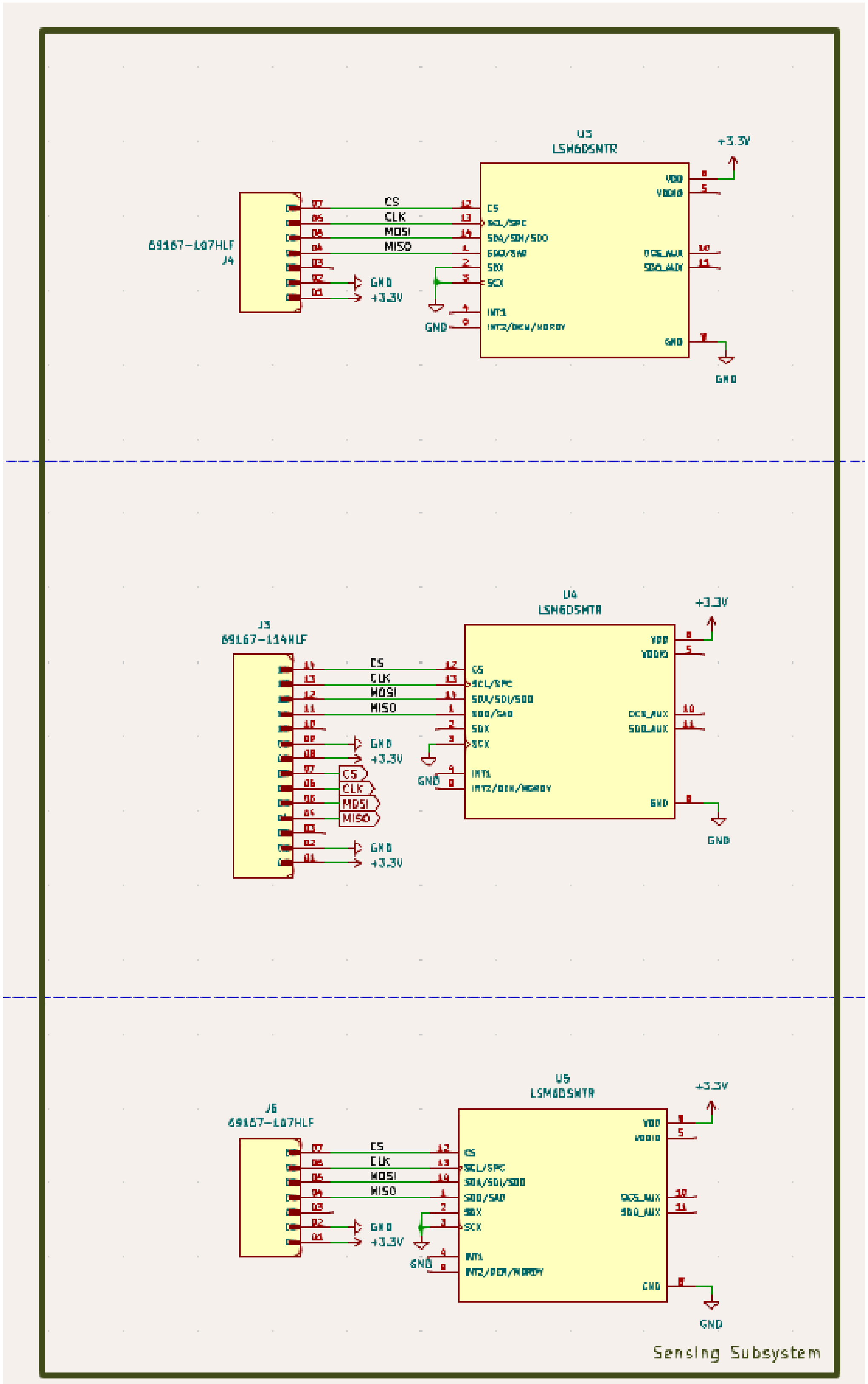
Requirement

Each IMU sensor should provide data at a rate of at least 20 Hz for a total combined rate of 60 Hz

Verification
Equipment
Precision Dumbbell Assistant, Oscilloscope
Test Procedure
1. Connect two sensor boards to the main board as shown in the visual aid 2. Connect the oscilloscope to a GPIO pin on the ESP32 3. Write a test program that toggles the value of the GPIO pin after receiving data from each of the three sensors. 4. Check the frequency of the GPIO pin. If it is higher than 10 Hz, then the test is successful. 10 Hz is the target value because the pin is toggled on every read, so there will be two reads for every period of the GPIO signal.
Presentation of Results
Record the results of the test as a single value in the notebook and mark whether the test passed or not.

Requirement
Each sensor board and accompanying wire harness should weigh less than 50g
Verification
Equipment
Sensor Board and Wire Harness, Scale
Test Procedure
1. Place the sensor board and wire harness on the scale, and record the value. 2. If the board and harness weigh less than 50g, the test is successful.
Presentation of Results
Record the results of the test as a single value in the notebook and mark whether the test passed or not.

Circuit Schematics:



2.3.2 Processing Subsystem

The processing subsystem consists of two ESP32 microcontrollers, which are responsible for reading in and processing the raw data from the sensing subsystem. Each microcontroller is connected to three sensors using SPI. We will use one SPI peripheral on each ESP32, and 3 GPIOs as chip selects to select which IMU we are talking to. The ESP32s will request data from each sensor one at a time. Then, they will use the raw data they receive to calculate the new position and orientation of each sensor. Finally, they will run the new coordinates through an algorithm to determine if the user's arms are in the correct position or not. This subsystem will also be responsible for the startup sequence used to calibrate the sensors. It will use the feedback subsystem to notify the user that they should hold their arms in a certain position (or a series of positions), and then use that data to get a base position in space for each sensor. The microcontrollers will also be connected to the buzzers in the feedback subsystem using PWM. They are also part of the wireless communication subsystem, which will be discussed later. Finally, the microcontrollers receive 3.3V from the power subsystem. The processing subsystem is related to two of our high-level requirements. More details can be found in the table below.

Requirement & Verification Tables:

Requirement
The microcontroller should be able to calculate the position and orientation of each sensor with an accuracy of ±5%
Verification
Equipment
LSM6DSMTR IMU sensor, ESP32 development board, Yardstick, Protractor
Test Procedure
Position:

1. Connect IMU sensor to ESP32 development board using manual.
2. Place the IMU sensor on a hard flat surface with the chip facing up, and a yardstick next to it.
3. Move the sensor along the yardstick 50cm.
4. Check to see if the calculated position has changed in the correct direction 50cm ±5%.
5. Repeat steps 3-4 5 times, and record the results
6. Repeat steps 2-5 for the Y and Z directions (stand the yardstick up for Z)
7. If all values are within the ±5% tolerance, the test passes.

Orientation:

1. Connect IMU sensor to ESP32 development board using manual.
2. Place the IMU sensor on a hard flat surface with the chip facing up, and a protractor next to it.
3. Using the protractor, move the sensor until it is 45 degrees from flat.
4. Check to see if the calculated orientation has changed in the correct direction 45° ±5%.
5. Repeat steps 3-4 5 times, and record the results
6. Repeat steps 2-5 for the other two axes (lay the protractor down for yaw)
7. If all values are within the ±5% tolerance, the test passes.

Presentation of Results

Record all readings in separate data tables in the notebook for the position and orientation, and take note whether each test passes or not.

Requirement
The orientation and position of each sensor should be calculated and analyzed at at least 20 Hz
Verification
Equipment
Precision Dumbbell Assistant, Oscilloscope
Test Procedure
1. Connect two sensor boards to the main board as shown in the visual aid 2. Connect the oscilloscope to a GPIO pin on the ESP32 3. Write a test program that toggles the value of the GPIO pin after every processing cycle (position and orientation update / analysis) 4. Check the frequency of the GPIO pin. If it is higher than 10 Hz, then the test is successful. 10 Hz is the target value because the pin is toggled on every cycle, so there will be two cycles for every period of the GPIO signal.

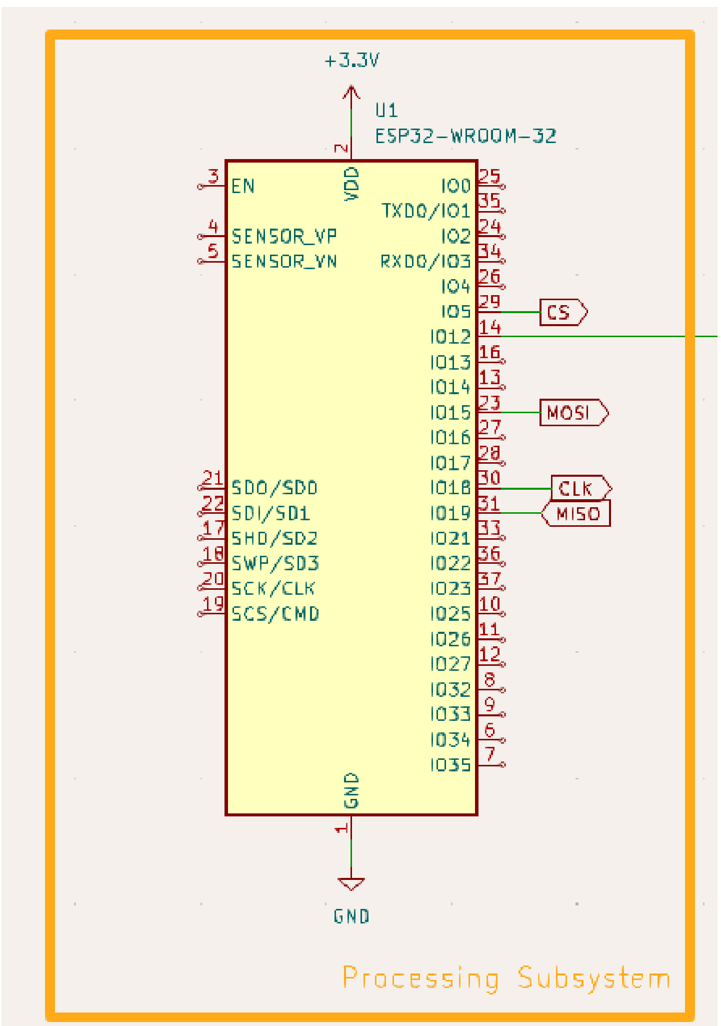
5. Run the test 5 times, and ensure that it meets the 20 Hz benchmark for every test.
Presentation of Results
Record the results of each test in the notebook and mark whether the test passed or not.

Requirement
The calibration sequence must take less than 20 seconds and provide consistent results.
Verification
Equipment
Precision Dumbbell Assistant, Stopwatch
Test Procedure
1. Put on the device, start the calibration sequence, and start the stopwatch 2. If the user is able to complete the calibration in 20 seconds or less, this part of the test passes 3. The user should perform 10 of the same correct movements to the best of their ability, and if they receive positive feedback on 9/10, the second test passes 4. Repeat steps 2-3 5 times, and if all 10 tests pass, the calibration sequence is verified.
Presentation of Results
Record the calibration time and number of correct movements for each trial in the notebook, and mark whether each test passed or not.

Requirement
SPI communication must operate between each IMU sensor and the ESP32 microprocessor at a speed of at least 10 ± 1.0 MHz.
Verification
Equipment

Oscilloscope, frequency counter, LSM6DSMTR IMU sensor, ESP32 development board
Test Procedure
1. Connect the oscilloscope to the SPI clock pin of both the IMU sensor and ESP32. 2. Initialize IMU sensor to transmit data to ESP32. 3. Use the frequency counter to measure the frequency of the SPI clock pin. 4. If the frequency value is between 9 and 11 MHz, this test is successful. 5. Repeat steps 2-4 10 times to generate enough data.
Presentation of Results
Include a data table for the frequency readings in the notebook and final report that includes all 10 trials.

Circuit Schematics:



2.3.3 Wireless Communication Subsystem

The wireless communication subsystem will handle the communication between an Android smartphone and the two ESP32s. We will be using Bluetooth Low Energy (BLE) for this task, which should provide sufficient bandwidth and range [4]. This subsystem contains both the ESP32s and smartphone, and is responsible for establishing and maintaining a BLE link. Each ESP32 will be paired with the smartphone, and will send any data needed by the smartphone app, which we will get to later in the feedback subsystem. This subsystem is essentially what ties the processing subsystem and the phone part of the feedback subsystem together. Both ESP32 and Android have well documented Bluetooth stacks, so we don't anticipate having to do any major work to make the devices talk to each other successfully.

Requirement & Verification Tables:

Requirement
There must be a Bluetooth Low Energy connection between the ESP32 microcontroller and the user's smartphone at a distance of at least 5 meters.
Verification
Equipment
ESP32 microcontroller, smartphone, measuring tape
Test Procedure
1. Place the ESP32 microcontroller at a fixed location. 2. Initialize ESP32 with a program that sets up BLE.. 3. Use measuring tape to measure 1 meter distance from ESP32. 4. Use a smartphone to connect to the ESP32 microcontroller through bluetooth. 5. Record if connection is successful. 6. Use measuring tape to move back 1 meter further. 7. Repeat steps 4-6 until connection is unsuccessful. 8. If the final distance is at least 5 meters, this test is successful.
Presentation of Results

Include a data table recording the distances and connection success or failure in the notebook and final report.

Requirement
The BLE connection must have packet loss of 20% or less at 5m.
Verification
Equipment
ESP32 microcontroller, smartphone, measuring tape
Test Procedure
1. Place the ESP32 microcontroller at a fixed location. 2. Initialize ESP32 with a program that sends BLE notifications to a smartphone located 5m away at a set interval for 5 minutes 3. Measure how many messages are received on the smartphone out of how many are sent, and calculate the percentage of messages that were received. 4. Repeat this test 4 times in varying conditions (inside, outside, line of sight, etc.)
Presentation of Results
Record the results of the test in a table and label which results go with which condition and mark whether the test passed or not. This table should be in both the notebook and final report

2.3.4 Feedback Subsystem

The two ESP32 microcontrollers used in the processing subsystem will drive the feedback system. Each ESP32 will be connected to a buzzer through its PWM pin. When the processing subsystem determines that the user is using improper form, the ESP32 will turn on the buzzer to audibly alert the user. The PWM pin will allow us to control the pitch and duration of the buzzer. The ESP32 microcontrollers will also be connected to a phone app using Bluetooth Low Energy. The microcontrollers will send the processed data to the app where the interface will allow the user to visually see feedback on their form. The user will be able to see how many repetitions they have completed and how many of those repetitions utilized proper form. Ideally, we will create a graphic to show the proper form and how the user differs from said form.

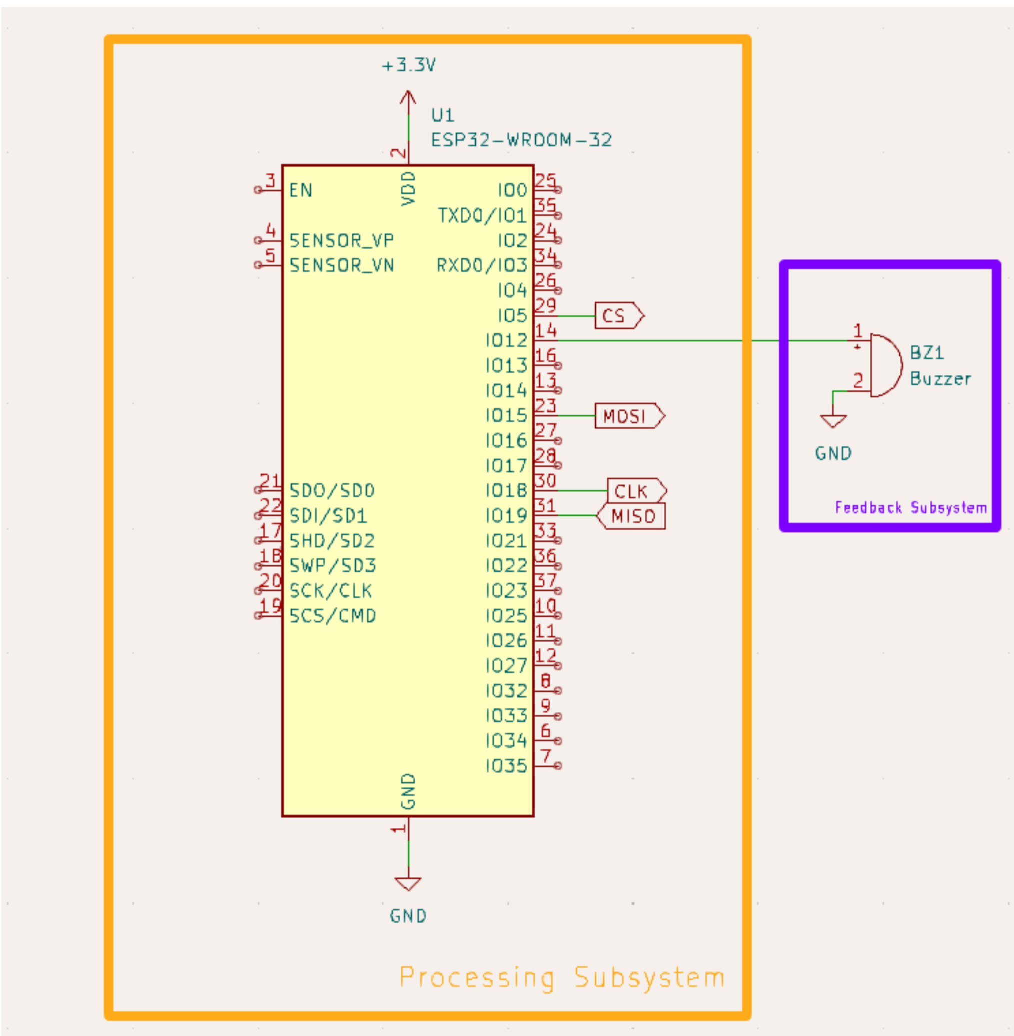
Requirement & Verification Tables:

Requirement
The buzzer must have a pitch between 200 Hz and 600 Hz with a tolerance of ± 10% and the sound must last at least 1 second.
Verification
Equipment
Buzzer, oscilloscope, function generator
Test Procedure
1. Connect the buzzer to the function generator using corresponding terminals. 2. Use the function generator to generate a square function with a frequency of 200 Hz. 3. Use the oscilloscope to record the frequency of the resulting sound. 4. Increase the frequency of the function by 100 Hz. 5. Repeat steps 3 and 4 until you reach 600 Hz. 6. If each recorded frequency is in the range of the input frequency ± 10%, this test is successful.
Presentation of Results
Include a data table recording the input and output frequencies in the notebook and final report. Also, plot a line graph with input frequencies on x axis and output frequencies on y axis.

Requirement
The buzzer must have a loudness of at least 60 dB
Verification
Equipment
Buzzer, function generator, smartphone

Test Procedure
1. Connect the buzzer to the function generator using corresponding terminals. 2. Use the function generator to generate a square function with a frequency of 200 Hz. 3. Use a smartphone decibel measurement app to measure the loudness of the buzzer. 4. Increase the frequency of the function by 100 Hz. 5. Repeat steps 3 and 4 until you reach 600 Hz. 6. If each recorded frequency has a loudness above 60 dB, the test passes.
Presentation of Results
Include a data table recording the loudness for each frequency, and note whether the test passes for each in the notebook and final report.

Circuit Schematics:



2.3.5 Power Subsystem

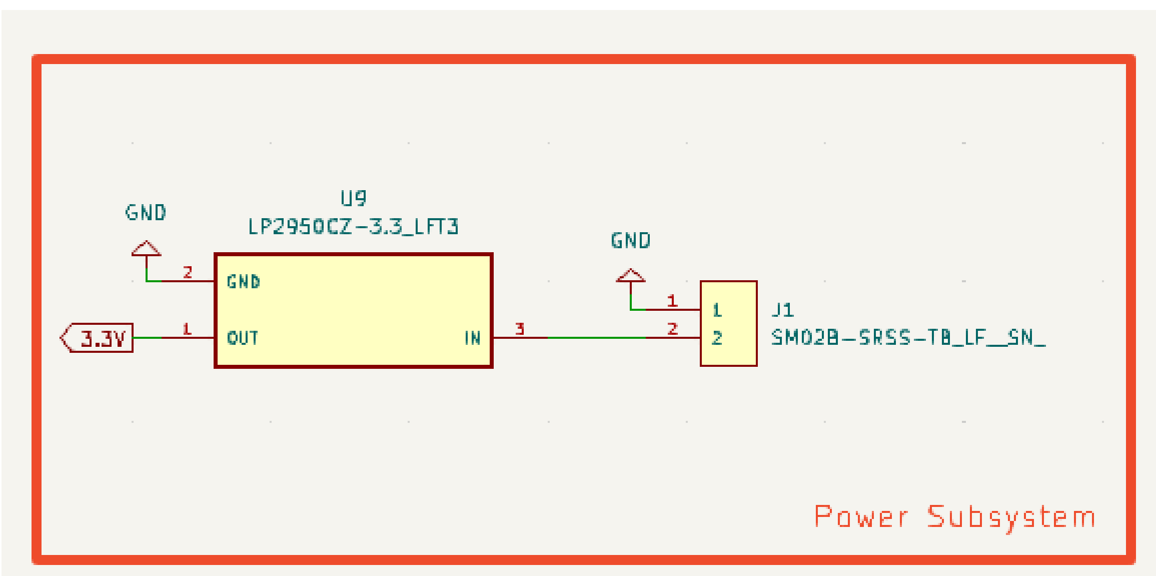
In the power subsystem, power will come from two 3.7V Li-ion battery packs, one on each arm. These will be located in an enclosure under the main board. We will use a LP2950CZ voltage regulator on each arm to drop the 3.7V of the battery pack down to the 3.3V needed to supply the sensors and microcontroller in the sensor and processing subsystems respectively.

Requirement & Verification Table:

Requirement
The voltage regulator must regulate the voltage to 3.3 V ± 5% in order to ensure that there is enough power supply to ESP32 and sensors but not enough power to damage electronic components. The LP2950CZ has internal protection circuitry including short circuit protection and thermal protection that will protect against many accidental issues.
Verification
Equipment
Digital multimeter, 3.7 V lithium batteries, voltage regulator prototype
Test Procedure
1. Connect the 3.7 V lithium batteries to the voltage regulator prototype using the corresponding terminals. 2. Make sure the batteries are on and supplying voltage. 3. Record the final voltage after the regulator using a digital multimeter. 4. Vary the resistance in the voltage regulator. 5. Repeat steps 3 and 4 10 times and record the results. 6. If each recorded voltage value is in range 3.3 V ± 5%, this test is successful.
Presentation of Results
Include a data table recording the voltages from all 10 trials in the notebook and final report. Also, plot a line graph with input resistance on x axis and output voltage on y axis.

Requirement
Each main board, battery, and enclosure should weigh less than 100g
Verification
Equipment
Main Board, Battery, Enclosure, Scale
Test Procedure
1. Place the assembled main board, battery and enclosure on the scale and record the value. 2. If the assembly weighs less than 100g, the test is successful.
Presentation of Results
Record the results of the test as a single value in the notebook and mark whether the test passed or not.

Circuit Schematics:



2.4 Tolerance Analysis

The critical feature that will be analyzed is the accuracy of the sensors to detect the user's motion. This is definitely the biggest risk to this project being a success since the user will receive incorrect feedback if the device is operating based on inaccurate sensors. Incorrect feedback can even lead to injuries for users. Both the MPU-6050 3-axis accelerometer and 3-axis gyroscope of the ESP32 microcontroller have limitations in their accuracy.

Hence, we will conduct a mathematical tolerance analysis to determine how much impact the worst case scenario results will have on detecting the form of the user. Let's assume that the accelerometer has an accuracy of ± 0.1 g (acceleration due to gravity) and the gyroscope has an accuracy of ± 0.01 dps (degrees per second) [3]. We will verify that the sample data is within the tolerance range of ± 5%.

Sample vs Expected Accelerometer Readings During Bicep Curls

Time (s)	Acc-x (g)	Acc-x (g)	Acc-y (g)	Acc-y (g)	Acc-z (g)	Acc-z (g)
	Sample	***Expected***	***Sample***	***Expected***	***Sample***	***Expected***
0.0	0.0	0.0	9.8	9.8	-0.0	0.0
0.1	-1.1	-1.0	9.3	9.3	-0.2	-0.1
0.2	-1.9	-2.0	8.7	8.8	-0.2	-0.2
0.3	-3.1	-3.0	8.2	8.3	-0.3	-0.3
0.4	-4.1	-4.0	7.9	7.8	-0.5	-0.4

Sample vs Expected Gyroscope Readings During Bicep Curls

Time	Gyr-x	Gyr-x	Gyr-y	Gyr-y	Gyr-z	Gyr-z

(s)	(dps)	(dps)	(dps)	(dps)	(dps)	(dps)
	Sample	***Expected***	***Sample***	***Expected***	***Sample***	***Expected***
0.0	0.00	0.00	0.00	0.00	0.00	0.00
0.1	0.09	0.10	0.21	0.20	-0.09	-0.10
0.2	0.21	0.20	0.39	0.40	-0.21	-0.20
0.3	0.31	0.30	0.59	0.60	-0.31	-0.30
0.4	0.39	0.40	0.81	0.80	-0.41	-0.40

The data tables above contain sample and expected data that can be read by the 3-axis accelerometer and 3-axis gyroscope while the user does bicep curls. We will now calculate the percent error for a few values to ensure that each value lies within the 5% tolerance range.

Let's use Acc-y value at 0.2 seconds below:

Percent error = (|8.7 - 8.8| / 8.8) * 100 = (0.1 / 8.8) * 100 = 1.14% < 5 %

Let's use Gyr-x value at 0.4 seconds below:

Percent error = (|0.39 - 0.40| / 0.40) * 100 = (0.01 / 0.40) * 100 = 2.50% < 5 %

Since the percent error calculations for both accelerometer and gyroscope are below 5%, we can conclude from the mathematical analysis that the readings are valid to guarantee proper form.

3 Cost & Schedule

3.1 Cost Analysis

Components:

Item Number	Designator	Qty	Manufacturer	Mfg Part #	Value	Description	Source	Unit Cost	Total Cost
1	U1, U2	2	HiLetgo	ESP-WROOM-32	N/A	Microcontroller	ECE Supply Center	$16.53	$33.06
2	BZ1, BZ2	2	AATC	AC-903-D-1P	3.3VAC	Piezo Buzzer	Mouser	$0.96	$1.92
3	U9, U10	2	Texas Instruments	LP2950CZ	3.3V	Voltage Regulator	ECE Services Shop	$1.09	$2.18
4	J1, J8	2	JST Sales America Inc	SM02B-SRSS-TB	N/A	Conn Header SMD	Digikey	$0.51	$1.02
5	N/A	2	TinyCircuits	ASR00012	3.7V	Lithium Ion Battery Pack	Mouser	$8.52	$17.04
6	N/A	4	Amphenol FCI	65039030LF	N/A	7 Pin Connector Cable	Electronics Service Shop	$0.92	$3.68
7	N/A	2	Amphenol FCI	65039027LF	N/A	14 Pin Connector Cable	Mouser	$1.23	$2.46
8	J2, J3, J6, J7	4	Amphenol FCI	69167-107HLF	N/A	7 Pin Connector PCB	Mouser	$1.44	$5.76
9	J4, J5	2	Amphenol FCI	69167-114HLF	N/A	14 Pin Connector PCB	Mouser	$2.07	$4.14
10	N/A	1	NTE Electronics Inc	WHS20-02-25	20 AWG	25 Feet 20 Gauge Wire	ECE Supply Center	$9.20	$9.20
11	N/A	1	Kester	24-6040-0039	N/A	040 60/40 ROSIN SOLDI	ECE Supply Center	$16.96	$16.96
12	N/A	1	Yuuro	N/A	N/A	12"x12"x1" High Density	Amazon	$6.99	$6.99
13	N/A	1	AnyCubic	N/A	N/A	1kg Spool PLA	Amazon	$14.99	$14.99
14	N/A	6	JNXQWE	N/A	N/A	Elastic Arm Bands	Amazon	$6.99	$41.94
									TotaL:
									$161.34

*Component cost does not include the cost of manufacturing the PCBs

Lab Equipment:

Item	Make	Model	Cost
Soldering Iron	Uline	H-10799	$190
Vise	Unkown	Unknown	~$20
Oscilloscope	Aigilent	DSO7104B	$24,081
Scale	Amazon Basics	Kitchen Scale with LCD	$12.59
Measuring Tape	Amazon Basics	Self-Locking Tape Measure	$7.97
Mutlimeter	Keysight	34461A	$1,457
3D Printer	Prusa	MK3	$899
Computer	PC	Unknown	~$1,000
Waveform Gene	Agilent	33500B Series	$3,973
			Total:
			$30,621

Labor:

To calculate labor cost we are assuming ~80 hours of work per person at $50 an hour for a total of $12,000.

Total Cost:

The total cost to complete this project, including the cost of parts, the cost of lab equipment, and the cost of labor but excluding the manufacturing cost for the 6 PCBs, is $42,783.34.

3.2 Schedule

Week	Task	Person
2/25 - 3/2	Start ordering parts for prototyping	Everyone
	Finish PCB schematic and start layout	Ellie
	Order sensor eval kit and start on sensor position and orientation calculations, make modifications to 3D printed enclosure	Cole
	Research BLE and get successful communication between dev board and computer	Ronit
3/3 - 3/9	If layout finished and audited, order PCBs, otherwise finish layout	Ellie
	Establish SPI communication between eval kit sensor and dev board	Cole
	Start prototyping Android app	Ronit
3/10 - 3/16	Spring Break	Everyone
3/17 - 3/23	Order PCB, start weighing components to ensure that they meet the weight limit, test standalone components like buzzer	Ellie
	Test sensor accuracy, test position and orientation calculation accuracy, start on form analysis algorithm	Cole
	Establish BLE communication between ESP32 and smartphone	Ronit
3/24 - 3/30	Assemble PCB, print battery enclosure	Ellie
	Develop calibration sequence, continue on form analysis	Cole
	Start developing app UI with dummy data, work out bugs with BLE	Ronit
3/31 - 4/6	Second PCB revision	Ellie
	Have working prototype for form analysis, have interface with feedback subsystem working	Cole
	Continue developing smartphone app	Ronit
4/7 - 4/13	Final PCB revision	Ellie
	Continue optimizing form analysis algorithm	Cole

	Finish smartphone app	Ronit
4/14 - 4/20	Fix remaining bugs	Everyone
4/21 - 4/27	Demo	Everyone

4 Ethics & Safety

4.1 Ethical & Safety Issues

Accuracy of Form: The most important ethical issue is obviously the accuracy of form enforced by our device as it could lead to injuries if inaccurate. To ensure this, we will use data from a certified online training source that we have referenced below. The IEEE and ACM code of ethics mention to prioritize user safety and having a risk of injury goes directly against the code [5] [7].

Privacy of Data: The privacy of data is also a cause of ethical concern as many people who workout are sensitive about information in their fitness diaries and quality of form in exercises. We will need to ensure safe storage of our data. The main safety issue is the overheating of batteries, which is why we will carefully check our voltage regulation. The IEEE and ACM code of ethics clearly state to prioritize privacy and confidentiality of user data [5] [7].

Wearable Device: We also have to make sure that the device is wearable for the safety of the user. The wires can't be too tight and should not prevent the user from moving their body parts naturally. We will have to carefully connect the microcontroller and sensors to make sure that the connections are not too stiff.

Safe Materials: Lastly, the materials that we use must be considered safe to contact human skin. Since voltage will travel through the materials used for the connections, they can't get overheated to the extent that it causes skin irritation.

Battery Safety: Lastly, the 3.7V batteries must not get overheated since they will be present in an enclosure that will be in very close contact with the user's skin. Overheating could also damage other electronic components in the design.

4.2 Safety Procedures

Accuracy of Form:

1. Refer to Verywell Fit source referenced below to generate ideal accelerometer and gyroscope readings for proper form for dumbbell bicep curls [6].
2. Perform verification procedure to make sure that the sensors are reading correct values while the user performs bicep curls.
3. Repeat step 2 regularly to make sure the device remains safe to use.

Privacy of Data:

1. When programming software, require the user to make a personal account on the application so that user needs to enter a personal password to access data
2. Generate a transcript for a user agreement in which the user is ensured that their data privacy is protected and that they consent to use and store their data on the application.

Wearable Device:

1. When placing the ESP32 microcontroller and IMU sensors, make sure that they are placed in a manner that they don't obstruct natural user movement (eg. bending elbows).

2. Try to use longer wires when connecting microcontrollers and sensors so that they don't obstruct user movement.

Safe Materials:

1. Verify that the materials being used in the design are in compliance with ACM and IEEE code of ethics.
2. Monitor the temperature of each subsystem during testing to make sure that no component is overheating.

Battery Safety:

1. Verify that the 3.7 V batteries being used are high-quality and are not prone to overheating.
3. Monitor the temperature of the batteries regularly during testing.

5 Citations

[1] HROSM, "Importance of Proper Weight Lifting Form - HROSM," Hampton Roads Orthopaedics Spine and Sports Medicine, Apr. 01, 2023. https://www.hrosm.com/importance-of-proper-weight-lifting-form/

[2] "LITHIUM ION POLYMER BATTERY 3.7V 1000MAH." Accessed: Feb. 22, 2024. [Online]. Available: https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/734/ASR00012_Web.pdf

[3] "iNEMO inertial module: always-on 3D accelerometer and 3D gyroscope." Accessed:

Feb. 22, 2024. [Online]. Available:
https://www.mouser.com/datasheet/2/389/lsm6dsm-1849628.pdf

[4] "Bluetooth LE & Bluetooth — ESP-FAQ documentation," espressif-docs.readthedocs-hosted.com. https://espressif-docs.readthedocs-hosted.com/projects/esp-faq/en/latest/software-framework/ble-bt.html#:~:text=The%20maximum%20throughput%20of%20Bluetooth (accessed Feb. 23, 2024).

[5] "IEEE Code of Ethics." *IEEE*, www.ieee.org/about/corporate/governance/p7-8.html. Accessed 22 Feb. 2024.

[6] Rogers, Paul. "Build Your Biceps with Dumbbell Curls." *Verywell Fit*, Verywell Fit, 7 Nov. 2022, www.verywellfit.com/how-to-do-the-biceps-arm-curl-3498604.

[7] "Software Engineering Code - ACM Ethics." *ACM Ethics - The Official Site of the Association for Computing Machinery's Committee on Professional Ethics*, 8 June 2022, ethics.acm.org/code-of-ethics/software-engineering-code/.