

Home Fitness Aid

Team 20: Andrew Garcia (ajgarci6@illinois.edu)
Hemanth Ravi Gowda (hr2@illinois.edu)
Steve Cheng (scheng40@illinois.edu)

Final Report for ECE 445, Senior Design, Fall 2020
TA: Sowjanya Akshintala

Table of Contents

Abstract.....	3
1 Introduction.....	4
1.1 High-Level Requirements.....	4
1.2 Visual Aid.....	5
1.3 Block Diagram.....	6
1.4 Features.....	7
2 Design.....	8
2.1 Control Unit.....	8
2.1.1 Design.....	8
2.1.2 Verification.....	9
2.2 Power Unit.....	10
2.2.1 Design.....	10
2.2.2 Verification.....	11
2.3 Sensor Unit.....	13
2.3.1 Inertial Measurement Unit.....	13
2.3.2 Calibration and Form Tracking.....	14
2.3.3 Verification of Sensor.....	16
2.3.4 Verification of Repetition Tracking.....	17
2.4 User Interface.....	18
2.4.1 LCD Display and Verification.....	18
2.4.1 Graphical User Interface.....	19
3 Costs.....	21
3.1 Cost Analysis.....	21
3.1.1 Labor.....	21
3.1.2 Parts.....	21
3.1.3 Grand Total.....	22
4 Conclusion.....	23
4.1 Ethics Considerations.....	23
4.2 Future Work.....	24
5 References.....	25

Abstract

The Home Fitness Aid is a wearable device responsible for taking measurements relating to the completion of repetitions of exercises such as bicep-curls or sit-ups. The device will then use this data to give the user advice on how to improve their form while doing that exercise. A progression system that increases the difficulty of exercises over time is also included. The device was successful in meeting these requirements and data is included within this paper to prove that claim. Furthermore, data relating to various subsystems such as the power unit, sensor, and microcontroller is also included.

1 Introduction

Across the world, governments are increasingly concerned about the health of their citizens. One issue affecting the health of a population is obesity, which can increase risk for diabetes, heart disease, and cancer [1]. Furthermore, the burden put on society due to obesity is \$147 billion in the United States alone [2]. According to the Centers for Disease Control(CDC), the best way to combat obesity and improve health is to increase physical activity through an exercise routine [3]. One of the greatest difficulties in starting an exercise routine is learning exercise form and counting repetitions of exercises. An improper exercise can risk injury to the individual and others. Additionally, without a system to keep track of the number and type of exercises performed, it is difficult to gauge the success of a workout plan over time. The combination of proper and regimented exercise is essential to succeed in weight management and healthy living.

The Home Fitness Aid is a wearable that can track exercises qualitatively and quantitatively. The device consists of an accelerometer, a gyroscope, and two microcontrollers. The accelerometer will track the speed, time, and direction of exercises done and the gyroscope was not used in the final implementation of the project. The accelerometer will process its data through a microcontroller which will then output its data via Bluetooth to a computing unit. This unit processes the data from the wearable to create data about the form of an exercise and the number of repetitions of an exercise. The data will then display on an LCD touchscreen for the user. The user can then use this data to track and change their exercise routine. Ultimately, the device succeeded in all of these aspects. Data proving the functionality of the device as well as the functionality of its components is included in this paper.

1.1 High-Level Requirements

High-Level Requirements:

- The counting of repetitions must be precise to an error rate of 10%.
- The product must be able to dynamically increase the difficulty of the exercise routine by increasing the repetitions or time. This occurs when the user completes three successful consecutive repetitions by an error rate of 10%.
- The product must be able to point out suggestions to the user so that he/she can perform the exercise correctly. When over 10% of the repetitions in a set deviate from the mean of the correct form, a suggestion will be prompted on the display.

1.2 Visual Aid

The physical design, shown in figure 1, consists of a wearable that is meant to be worn on the arm or leg, and a separate screen for the user interface.

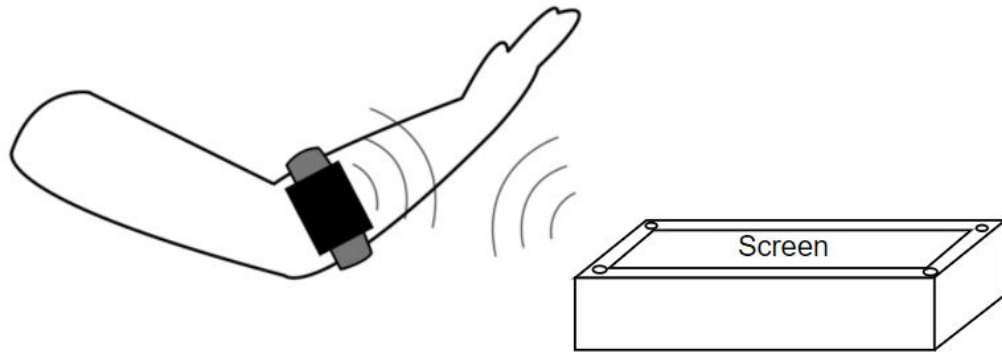


Figure 1: Sketch of Proposed Product

1.3 Block Diagram

To meet its operational requirements the device is split into four units: a sensor unit, a power unit, a control unit, and a user interface. The power supply ensures that the device is powered for the duration of a few exercise sessions estimated at an hour every other day. The sensor unit contains an accelerometer and gyroscope that measures the angular velocity and linear acceleration respectively. This information is then sent to the control unit which is responsible for processing and outputting the exercise data to the user interface. Furthermore, the control unit can also receive commands from the user interface to enter a new exercise, calibrate, etc. Both microcontrollers communicate via built-in Bluetooth antennas. The SPI interfaces will handle the data transfer between the microcontroller and the sensor/display.

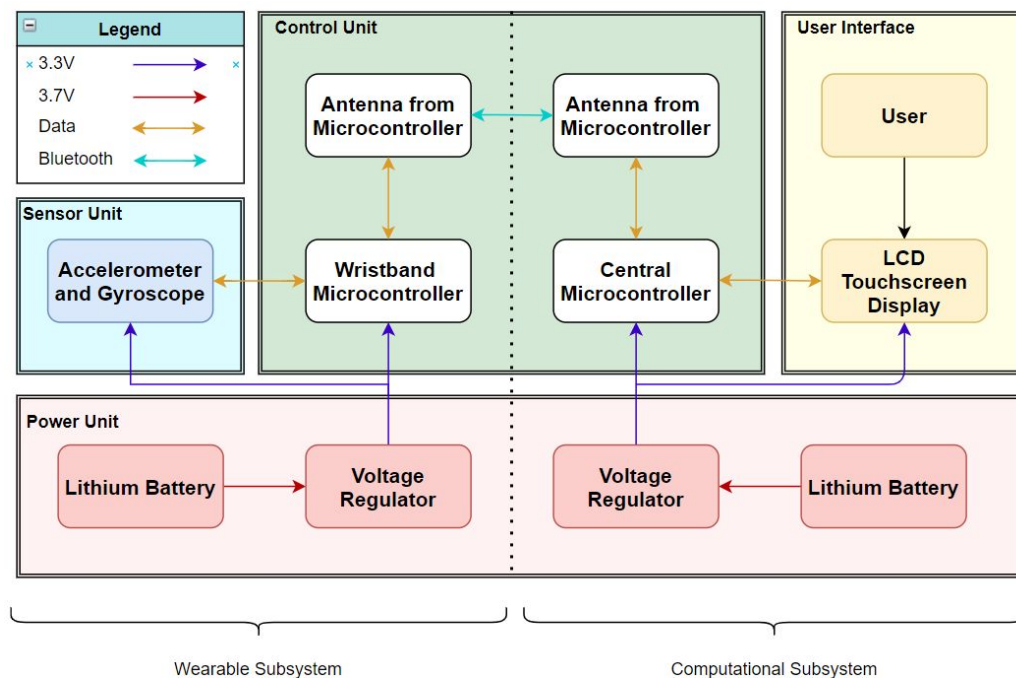


Figure 2: Home Fitness Aid Block Diagram

1.4 Features

There are five critical features implemented in the Home Fitness Aid to have a completed product. They are shown in table 1 below, followed by a small description.

Feature	Description
Custom Exercises	Users are able to enter in any exercises they wish. This will take the name of the exercise, followed by the number of sets and reps they wish to complete.
Calibration System	Takes the measurements for the custom exercises for the purpose of counting repetitions.
Pre-selected Exercises	A list of calibrated exercises for the user's selection.
Form Tracking	Notifies the user if their form has any deterioration that may reduce the quality of their exercise.
Dynamic Progression	Increases the difficulty of the routine when the user consistently completes several exercise routines.

Table 1: Features

2 Design

2.1 Control Unit

2.1.1 Design

This unit is responsible for relaying information from the sensors to the display while also processing the data received from the sensors to determine whether a repetition should be counted or not. Our requirement for our microcontrollers is that they can transmit and receive 90% of Bluetooth Low Energy (BLE) packets over a reasonable distance of 5 meters. This requirement was to significantly reduce the chance that the microcontroller would fail to detect more than 10% of the critical points when the user performs his/her exercise routines.

For our project we chose the ESP32-WROOM-32E module, which is a fully integrated BLE module. This allows us to treat the module like a microcontroller without having to separately connect the essential hardware components for wireless communication. There are various other alternative modules, such as the BR-LE4.0-S3A and the BLE113 modules, that the designer can choose from if he/she chooses to rebuild this device. The reason why we chose this device is because there are additional features in this module that are very likely to be used in our future designs, such as its low power sleep modes and its large amount of flash memory to store more user interface images for the display. Moreover, the ESP32 SoCs are very popular among embedded systems developers and thus there is more technical support available for beginners who wish to use such modules.

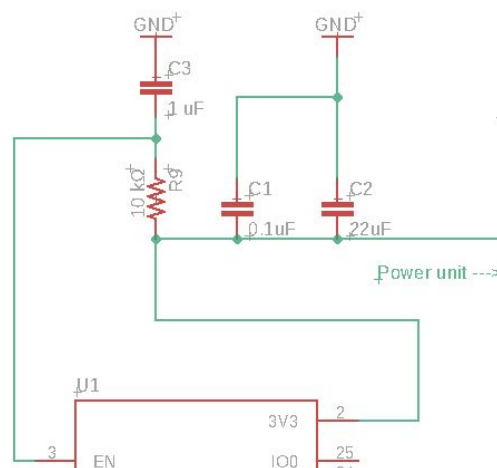


Figure 3: Microcontroller to power unit schematic

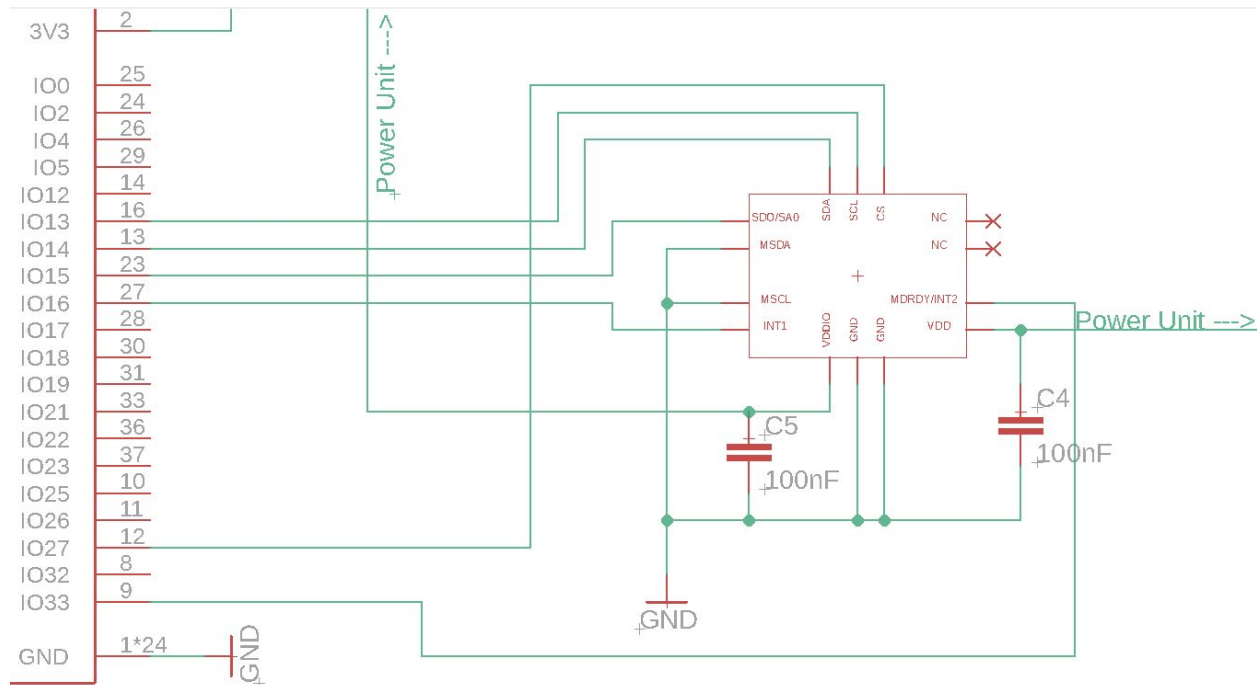


Figure 4: Microcontroller to sensor schematic

2.1.2 Verification

To verify our requirements, the client microcontroller will count the total number of packets received divided by the value of the last packet read. Our server microcontroller will send packets with an integer value. The value starts at 1 and gets incremented when the server microcontroller is about to send the next packet. For every packet received, the client microcontroller will output the results on our serial monitor. The program files can be found at the “References” section of our document. In addition, we have made sure that our microcontrollers were at least 5 meters apart.

As you can see in Figure 5, we satisfy our requirement of successfully transmitting and receiving at least 90% of the BLE packets.

```
Number of Packets Recieved:276.00
Expected Number of Packets:281
Success Rate:0.98
```

Figure 5: Screenshot of serial monitor output from client microcontroller

2.2 Power Unit

2.2.1 Design

This unit provides power for the entire device and will receive power from a battery. A lithium-ion battery was selected because of its low-cost, rechargeability, and commonality. For the wristband subsystem, the battery must be able to be used for an hour every other day for at least two days but also be small enough to fit into a wearable device. Since the wristband microcontroller and the sensor unit were estimated during design to require about 150 mA of current to run, a battery of 350 mAh was selected for the wearable. A low-cost charger was selected that was capable of charging the battery at a minimum of 100mAh. A larger battery would extend the operational time of the device but take longer to charge. Once the battery and battery charger was selected the power unit itself was designed. The final schematic for the power unit is shown in Figure 6 below.

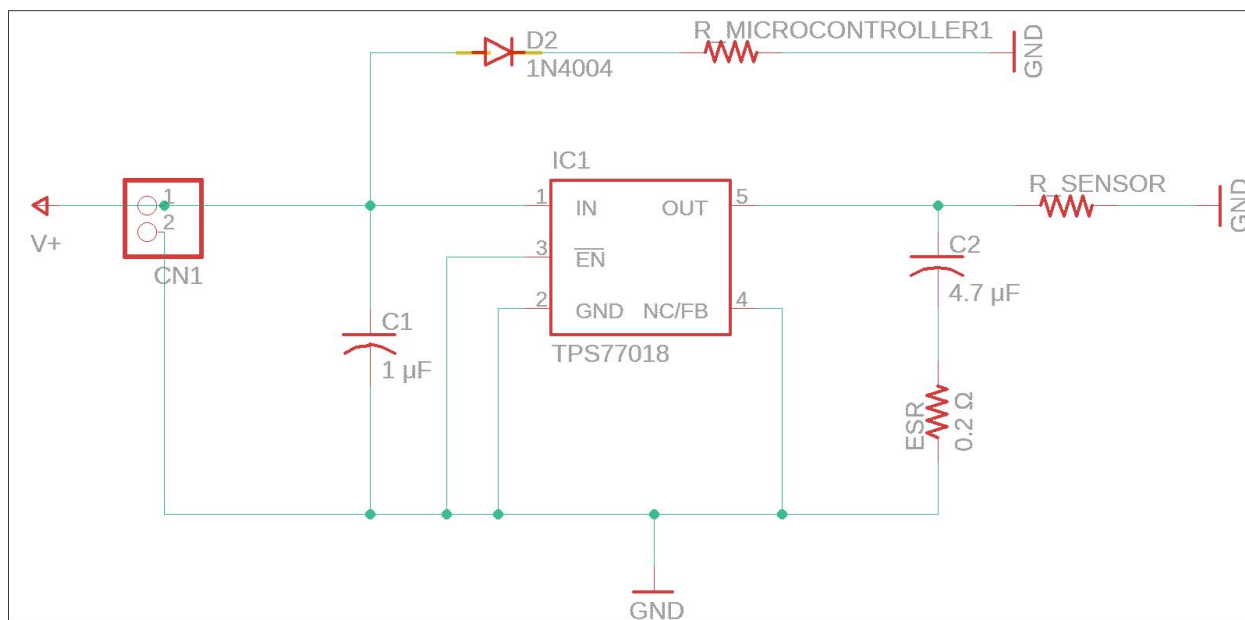


Figure 6: Power Unit Schematic

In this schematic CN1 was the PH-connector selected to connect the battery to the device. Once connected the battery would provide voltage at 4.2 V if fully charged and 3.7 V if near discharge. This voltage would then go through a 1N4004 diode or a Texas Instruments TPS77018 voltage regulator in order to supply the correct voltage to the microcontroller and sensor. A Texas Instruments voltage regulator was chosen to provide a voltage drop between the battery and sensor because of the large voltage drop required and the low current requirement of the sensor. Efficiency concerns due to the large voltage drop were disregarded due to the low current requirement of the sensor at 0.45 mA but was still measured. Replacing the regulator with a series of

diodes or resistors is possible but this solution would be more inefficient relative to the voltage regulator. In contrast, a 1N4004 diode was chosen to provide a voltage drop between the battery and microcontroller because the diode provides a constant 0.7 voltage drop and requires soldering less components on the PCB relative to a voltage regulator. The computational subsystem and the user interface were changed to be powered through a usb power supply after the failure of the computational subsystem PCB.

2.2.1 Verification

Lithium-Ion Battery

The requirements for the lithium-ion battery are as follows:

1. Provides a minimum of 350 mAh for wearable unit and sensor unit.
2. Provides a minimum of 2000 mAh for the computation unit.

Requirement #2 was met by choosing to power the computational subsystem through a USB power source. For requirement #1, a simple resistor circuit was set up and a fully charged battery was discharged at a current measured at 132 mA at full charge and 101 mA at discharge. This current decrease occurred linearly over time. This continued for 3.67 hrs until fully discharged and no current flowed. The proof for verification is included below:

$$\begin{aligned}
 Q_{min} &= 350 \text{ mAh} \\
 I &= 0.5 * (132 + 101) = 116.5 \text{ mA} \quad (1) \\
 t &= 3.67 \text{ hrs} \\
 Q &= I * t \quad (2) \\
 Q &= 116.5 * 3.67 \\
 Q &= 427.55 \text{ mAh}
 \end{aligned}$$

Since $Q > Q_{min}$, the lithium-ion battery is verified.

Voltage Regulator

The requirements for the lithium-ion battery are as follows:

1. Provides $1.8 \text{ V} \pm 10\%$ from a $3.7 \text{ V} - 4.2 \text{ V}$ source.
2. Can operate at a current of $\pm 10\%$ 0.45 mA.
3. Operates at an efficiency of 43% at max voltage and 49% at minimum voltage.

After the wristband PCB was soldered the voltage regulator was tested in the ECE lab by placing a probe on the output of the voltage regulator and measuring the voltage when a DC voltage of 3.7 volts and 4.2 volts is applied to the voltage regulator input. A voltage output of 1.8 volts was measured when 3.7 volts was applied and a voltage output of 2.3 volts was measured when 4.2 volts was applied. These results provide

verification for requirement 1. Furthermore, since the sensor was able to operate on the PCB board with the provided voltage and current requirement 2 was verified. Using an equation from the TPS77018 datasheet the power efficiency was calculated as follows [5]:

$$P_{VR} = (V_I - V_O) * I \quad (3)$$

$$P_{sensor} = V_O * I \quad (4)$$

$$\%_{eff} = P_{sensor} / (P_{sensor} + P_{VR}) * 100\% \quad (5)$$

$$I = 0.45 \text{ mA}$$

$$V_I = 3.7 \text{ V}, 4.2 \text{ V}$$

$$V_O = 1.6 \text{ V}, 2.0 \text{ V}$$

$$P_{VR1} = (3.7 - 1.8) * 0.45 \text{ mA} = 0.855 \text{ mW}$$

$$P_{VR2} = (4.2 - 2.0) * 0.45 \text{ mA} = 0.99 \text{ mW}$$

$$P_{sensor1} = 1.8 * 0.45 \text{ mA} = 0.81 \text{ mW}$$

$$P_{sensor2} = 2.0 * 0.45 \text{ mA} = 0.90 \text{ mW}$$

$$\%_{eff1} = 0.81 / (0.81 + 0.855) * 100\% = 49\%$$

$$\%_{eff2} = 0.90 / (0.90 + 0.99) * 100\% = 47.6\%$$

Since the resulting efficiencies are greater than or equal to their expected values the voltage regulator is verified.

Battery Charger

The requirements for the battery charger are as follows:

1. Charges Li-ion battery to 4.2-4.16 V when a continuous voltage of 4.4-7.0 V is applied.

For this experiment there is a green light on the battery charger which flashes once charging is complete. A USB power source is used to provide power to the charger at 5.0 V. A discharged battery was plugged into the battery charger and the green light was off. An initial voltage measurement of the battery was taken while plugged in at 4.02 V. After a period of time, the green light flashed and another measurement was taken at exactly 4.20 V and did not exceed this voltage when measurements were taken hourly after .

2.3 Sensor Unit

The inertial measurement unit(IMU) is the primary means of detecting the user's movements through their exercise routine. The chosen IMU for our product is the LSM6DSL from Texas Instruments. We have a Serial Peripheral Interface(SPI) between the sensor and the microcontroller. This unit contains an accelerometer and gyroscope which measures linear and angular acceleration respectively. The sensor will supply the necessary data for consistent tracking, and also for both the Calibration and Form Tracking System which will be discussed in the next section.

2.3.1 Inertial Measurement Unit

There are two possible forms of tracking that can be used in our device, angular tracking and spatial tracking. Ideally we can use a combination of both, but the complexity of spatial tracking negates itself for the purposes of our product. It will need at least another sensor, such as a camera, and much more time for implementation. This leaves angular tracking as the only way for tracking the orientation of the wearable device, namely the "Aircraft principle axis". This is the roll, pitch, and yaw angles of the wearable, as shown in figure 7 below. In our product, we will only be using roll and pitch. The yaw is incalculable with only a single accelerometer, as the equations use the gravity force vector as a reference point in calculating the angles. Since there is not a change in acceleration when we simply rotate the accelerometer, it won't be able to detect the yaw. Another sensor would be needed, such as a magnetometer to give another point of reference. Thankfully, most exercises have some sort of angular displacement for roll and pitch, so the yaw is not as important and can be implemented in our future works.

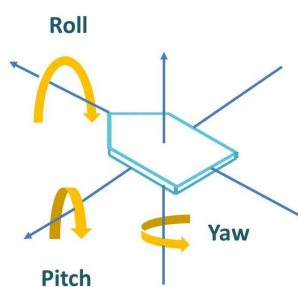


Figure 7: Aircraft Principle Axis

The roll and pitch of the wearable can be calculated with the equations shown in figure 8 below. Notice how only the accelerometer data is used here. We can safely drop the requirements of the gyroscope in our product, as the accelerometer can calculate nearly all the information that we need.

$$roll = \arctan\left(\frac{accelY}{\sqrt{(accelX^2 + accelZ^2)}}\right) * \frac{180}{\pi}$$

$$pitch = \arctan\left(-\frac{accelX}{\sqrt{(accelY^2 + accelZ^2)}}\right) * \frac{180}{\pi}$$

Figure 8: Equations for Pitch and Roll

2.3.2 Calibration and Form Tracking Systems

The calibration and form tracking systems provide the fundamental features of this product. The implementation of these systems are approached with a simple method to give reliable feedback to the user. The calibration of any custom exercises takes in the row and pitch of the relaxed and the exerted position of the exercise. The relaxed position is simply the starting point of the exercise. This can be taken when the user holds still in the starting position during the calibration period. After the initial position is taken, the user is then given approximately ten seconds to complete a single repetition. The exerted position is simply defined as the point in which the roll and pitch deviate most from the relaxed position during that single repetition. Then we can count a repetition of an exercise when the wearable can meet both the row and pitch of the relaxed and exerted position.

It may seem like the criteria for the count of a repetition is too easily satisfiable from the previous description. After all, a repetition can be counted even if the user's form is suboptimal. We introduce a simple way for the user to receive feedback when their form deteriorates. During the exercise routine, the form tracking system generates an array of points in between the relaxed and exerted position. A simple diagram illustrates this system in figure 9. We are making an assumption that most exercises will be linear, and will not have any quadratic-like behavior. We expect the user to meet these row and pitch points to count as a repetition done with a good form. If more than 10% of these points are missed, a suggestion is sent to the user to correct their form. There is a buffer for the angles of the row and pitch, so the user just needs to be within that buffer for it to count. The buffer is about ± 10 degrees from the generated points.

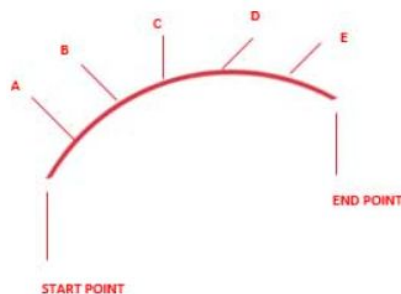


Figure 9: Illustration of Form Tracking

The form tracking system is one of the high-level requirements for this product. After extensive trials, we can safely say the software development of the form tracking system works well. Below in figure 10, there is a code piece from the form tracking system. For each repetition, it repeatedly checks the relaxed and exerted position matches with the current user's position. It also checks if the ten generated points are met. If over two of them are missed, then it sends a prompt to the user to improve their form.

```
if (currentRoll > exertedRoll.toInt()-8 && currentRoll < exertedRoll.toInt()+8)
    exertedRollMatched = true;
if (currentPitch > exertedPitch.toInt()-8 && currentPitch < exertedPitch.toInt()+8)
    exertedPitchMatched = true;

//Check if the pitch and roll in the middle of an exercise is met. Used for form control.
for (int i=0; i<10; i++){
    if(currentRoll > linearPositionRoll[i]-12 && currentRoll < linearPositionRoll[i]+12 && cu
        linearPositionCheck1[i] = true;
}
```

Figure 10: Piece of Form Tracking Code

2.3.3 Verification of Sensor

There are two requirements that must be fulfilled to ensure smooth tracking of the wearable. The following are the requirements for the accelerometer:

- The inertial measurement unit must sample data at a rate of at least 50 measurements per second.
- Accelerometer must have a precision of at least $\pm 1.0 \text{ m/s}^2$.

To test the data sampling rate of the accelerometer, we simply wrote a piece of code that counts the number of times the accelerometer data can be sampled in a single second. We found that the sampling rate is measured at an average of 12,160. When we introduced the calculations of the roll and pitch, and the transmission of data through Bluetooth, we average at about 4950 data samples per a second. A screenshot of the code's output is provided in figure 11.

```
Data Sampling Rate: 4950
Data Sampling Rate: 4951
Data Sampling Rate: 4953
```

Figure 11: Screenshot of Accelerometer Sampling Rate

For the second requirement, we can simply test the precision of the accelerometer when it is laying at rest. We expect the accelerometer to have an acceleration of zero in the x and y axis, but approximately 9.81 m/s^2 for the z axis due to the effects of gravity. The accelerometer data screenshot in figure 12 is taken when the unit is laid at rest on a tabletop. We have a slight deviation from the ideal standard, but it is well under the range of the requirement of $\pm 1.0 \text{ m/s}^2$.

```
Accelerometer:
  X = 0.2250
  Y = -0.0527
  Z = 9.9192
```

Figure 12: Screenshot of Accelerometer Data

2.3.4 Verification of Repetition Count

A high-level requirement that defines our product is the accuracy of the repetition tracker. Ideally, we wish for all custom exercises to work flawlessly, but we do have some limitations to our current device. For the following exercises in table 2, the exercises are performed with the wearable device strapped onto the wrist of the user except for one. For the exercises that have the most angle displacement, we notice that the counter works fantastically. The main limitation we have is when there are no angle changes to the exercises. For example, squats are difficult to track because the user moves vertically with virtually no arm motion. Since our device currently only tracks the roll and pitch angles, it has a very difficult time detecting those movements. The exercise's relaxed and exerted are nearly the same because the wearable is strapped onto the arm which does not change in angle. If the squats are completed with the wearable strapped on the calf, we can see much improved results since the angles are utilized. In the future, it's very possible to use the accelerometer to categorize exercises so the correct tracking can be implemented without unstrapping and moving the wearable device.

Exercise Name	Expected Count	Actual Count	Error Rate
Bicep Curls	20	20	0%
Sit-ups	20	20	0%
Push-ups	10	9	10%
Squats(Wrist)	20	28	29%
Squats(Calf)	20	19	5%

Table 2: Error Rate of Repetition Counting

2.4 User Interface

The goal of the User Interface(UI) is to provide the user with an intuitive and clean interface. The UI platform is hosted on a LCD touchscreen display which is the means of communication between the user and the product.

2.4.1 LCD Display and Verification

The chosen LCD display is a 320x240 TFT Touchscreen with a ILI9341 driver. We chose this display for it's touch capabilities and the appropriate size for our product. The screen must be large enough for clear readability and small enough for portability. The interfacing between the display and microcontroller is with the Serial Peripheral Interface(SPI).

We require the display to have the following requirements:

- Active screen area must be at least 40 x 50 mm large.
- The pixels per inch (PPI) must at least be 100 for readability.
- Touchscreen must be functional and accurate within an error of ± 8 pixels.

The first test is a simple measurement of the pixelated area in the display. The requirement is to be at least 40 x 50 mm for readability. In figure 13, the measurements are about 45 x 59 mm.

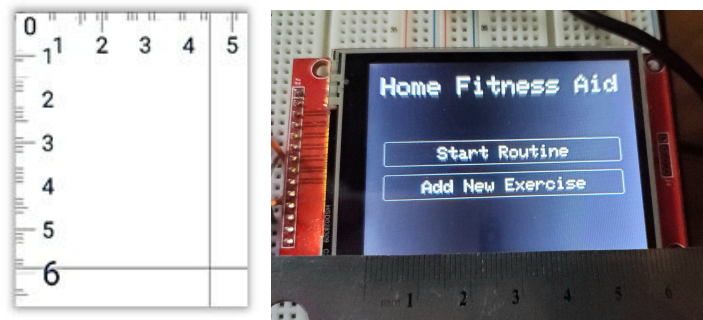


Figure 13: Measurements of display

For the second test, the requirement is to have at least 100 pixels per inch (PPI). The length of one of the sides is 59 mm or 2.32 inches. For the length of the display, there are 320 pixels. Then 320 pixels divided by 2.32 inches gives us about 138 PPI which is greater than the requirement. This will ensure readability.

The third and final test is the touchscreen must provide accuracy within an error rate of 8 pixels. In figure 14, the pixel at $x=200$ and $y=100$ was touched and the given result

was $x=204$ and $y=98$. Using Pythagorean theorem, we get that the distance between the actual pixel and the output pixel is about 4.47 pixel distance away. This is within the requirement of 8 pixels.

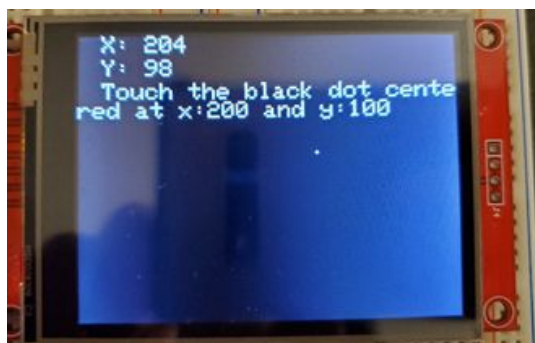


Figure 14: Accuracy test of touchscreen

2.4.2 Graphical User Interface

The user interface can be categorized into three functions: the menu screens, custom-exercise inputs, and the parts which utilizes the sensor's data.

The menu screens have a simple UI for quick exercise sessions with the product. When the device is first being used, new exercises must be added for the user to start their routine. There is a choice of pre-selected exercises and also any custom exercises. If the custom exercise is chosen, the user is first prompted for the name of the exercise which can be entered in the fully loaded keyboard. Afterwards the user can choose whether the exercise is a sets/ reps or a timer-based exercise. They can choose the difficulty to start with, which will increase with the succession of their routines. If the exercise is based off of sets/ reps, then the user will be prompted with the Calibration Screen with instructions. This part will take in the measurements needed for the counting of reps and form tracking. One of the high-level requirements is that the difficulty must increase after three successful and consecutive routines are complete. This means that if the user is able to complete the routines without pressing the "back" button and there are no new exercises added, then the user's routine increases in difficulty. With several tests, this is always guaranteed to occur when the criteria is met. In other words, the progression system in the high-level requirement is satisfied.

After the exercises have been submitted, the user can finally begin their exercise routine. Upon pressing "Start Routine" in the home menu, the UI begins to iterate over all the exercises in the order that it was entered. It will begin counting the reps when the criteria has been met. There is also a built-in break between each set and also each

exercise which can be modified. All the exercises and progression data are saved via the onboard flash memory for future use, even when power is lost.

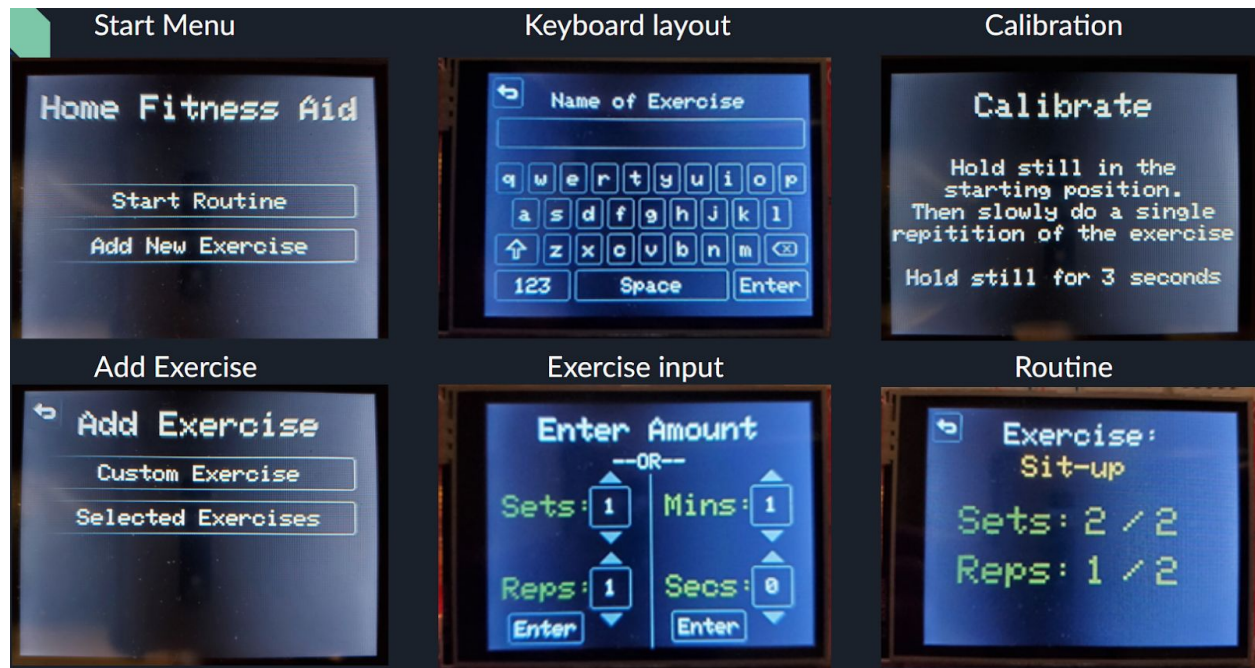


Figure 15: Screenshots of the User Interface

3 Costs

3.1 Cost Analysis

3.1.1 Labor

The main factor to the project's cost is labor. A graduate from UIUC's ECE program typically starts with an average salary of \$88,000, or approximately \$42/hr [6]. The design, manufacturing, testing, and documenting of the product took approximately 14 weeks. Our team contributed approximately 15 hours per a week per person. The table below summarizes the costs.

Name	Hourly Rate	Hours	Overhead Costs	Total
Andrew Garcia	\$42.00	210	x2.5	\$22,050
Hemanth Ravi Gowda	\$42.00	210	x2.5	\$22,050
Steve Cheng	\$42.00	210	x2.5	\$22,050
Machine Shop	\$56.12	6	x1.0	\$336.70
Total Labor Costs				\$66,486.70

Table 3: Labor Costs

3.1.2 Parts

Qt.	Part #	Manufacturer	Description	Price	Total
2	ESP32-WR OOM-32E	Espressif Systems	Dual-core 32-bit microprocessor with 4 MB SPI flash, SPI interface, Bluetooth LE, and Wi-Fi capabilities	\$2.80	\$5.60
1	ILI9341	HiLetgo	2.8" SPI TFT LCD Display with resistive touchscreen capabilities	\$13.99	\$13.99
1	LSM6DSL	STMicroelectronics	Accelerometer and gyroscope sensors	\$3.98	\$3.98
2	TPS77018	Texas Instruments	Voltage regulator	\$1.14	\$2.28
1	2750	Adafruit Industries	Lithium Battery 3.7V 350mAh	\$6.95	\$6.95
1	2011	Adafruit Industries	Lithium Battery 3.7V 2000mAh	\$12.50	\$12.50
1	1904	Adafruit Industries	Lithium Ion Polymer Charger Board	\$6.95	\$6.95
Total of all parts:					\$52.25

Table 4: Parts Costs

3.1.3 Grand Total

Table 5 below summarizes the total costs of labor and parts required for the completion of this product.

Section	Costs
Labor	\$66,486.70
Parts	\$52.25
Grand Total	\$66,538.95

Table 5: Total Cost

4 Conclusion

The senior design project for this semester has been particularly fast-paced and challenging. We encountered a great deal of frustration and issues, but ultimately, we walked away with more knowledge and experience. We have learned many aspects that go into a project, from the technical insights and designs, to how to function and work together as a team. As our project reaches its end, we can happily say that our product was a success. We met all the high-level requirements and even the requirements in each subsystem. Of course, there are still plenty of improvements that can be made if we wish to continue the project, but we are satisfied with our accomplishments during this semester.

4.1 Ethical Considerations

There are some common safety hazards in the development and usage of this product. This section will address these safety hazards and our plans to avoid any dangers. We will also address any ethical concerns that could result from the daily use of the product. The IEEE Code of Ethics will be used as a standard to judge any ethical and safety hazards that may emerge.

The first issue we will address is the safe usage of lithium batteries. Lithium batteries present a fire and explosion hazard if they are damaged or not properly managed. We will ensure in our design that the battery will be held in a safe location, and is not susceptible to small falls. Furthermore, we will take precautions to store the device in an environment where the lithium batteries are not subject to temperatures above 45 degrees Celsius or below 0 degrees Celsius where there is a possibility of thermal runaway and explosion. Insulating material will be extensively used to ensure short circuiting is not possible. Extensive testing of the circuitry will also occur to ensure the battery is not subject to voltages above its tolerances which can lead to cathode breakdown and a release of thermal energy. To properly charge the lithium battery, we plan to use Adafruit's battery charger at the recommended current [7]. Since we are also using Adafruit's batteries, the charging combination is designed to be stable. Lastly, IEEE Code of Ethics #1 states that one must "disclose promptly factors that might endanger the public or the environment" [4]. Since lithium batteries are a known environmental risk, we must ensure the used batteries are disposed of in a proper manner. We will also inform users of the device that it contains lithium batteries and is a hazard to the environment if not disposed of correctly.

4.2 Future Work

There exists several possible improvements to the system we have designed:

- Implementation of a spatial tracking system to track linear acceleration, linear velocity and 3D coordinates of the wearable device
- Use of low power mode on the microcontroller to reduce power consumption when BLE signals are not received or transmitted
- Reducing the size of the PCB components and lowering the size of the PCB itself to make the wearable feel more comfortable
- Using a flex PCB and replacing rigid parts such that the wearable device feels more comfortable
- Use of omni-directional antennas to improve BLE communication
- Implementation of a contention window to allow communication for multiple wearable devices

5 References

- [1] *Health and development?*, World Health Organization. Accessed on: Sept 9, 2020. [Online]. Available: <https://www.who.int/hdp/en/>.
- [2] *Adult Obesity Causes and Consequences*, Centers for Disease Control, 2020. Accessed on: Sept 9, 2020. [Online]. Available: <https://www.cdc.gov/obesity/adult/causes.html>.
- [3] *Benefits of Physical Activity*, Centers for Disease Control, 2020. Accessed on: Sept 9, 2020. [Online]. Available: <https://www.cdc.gov/physicalactivity/basics/pa-health/index.htm>.
- [4] *IEEE Code of Ethics*, IEEE, 2020. Accessed on: Sept 10, 2020. [Online]. Available: <http://www.ieee.org/about/corporate/governance/p7-8.html>.
- [5] Texas Instruments, "ULTRALOW-POWER 50-mA LOW-DROPOUT LINEAR REGULATORS," TPS77018 datasheet, June 1999 [Revised May 2001].
- [6] *Salary Averages*, University of Illinois. Accessed on: Sept 28, 2020. [Online]. Available: <https://ece.illinois.edu/admissions/why-ece/salary-averages>