PIKL (Paddle Integrated Kemper Logic)

A Technical Report submitted to the Department of Electrical and Computer Engineering

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Kemper Patrick Siever

Fall, 2024

Technical Project Team Members

Jiseoung Kim

Oscar Lauth

Wilmot Westriecher

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Adam Barnes, Department of Electrical and Computer Engineering

Table of Contents

List of Figures

List of Tables

Statement of Work

Jiseoung Kim: I was responsible for designing the PCB (printed circuit board) for the project. Before starting the design, I collaborated with Kemper to test the sensitivity of the piezoelectric sensors and assisted in developing the circuit. Once the circuit and PCB designs were completed, I assisted in populating the PCB with components. Since the PCB interfaces with several other components, I tested the connectivity of each one to ensure proper functionality. Finally, I assisted in integrating the overall hardware system into the paddle, ensuring consistent performance throughout the process.

Oscar Lauth: My primary responsibility was leading the software development for this capstone. I wrote the entirety of the embedded software running on the MCU which includes BLE service definitions and transmission, advanced ADC implementation for piezo sensor reading, IMU data fetching over I2C, and shot processing logic. I also assisted Wilmot with the GUI program, focusing on creating the BLE central on the Windows laptop in Python which received BLE data from the MCU. Further, I offered some consultation and assistance to the hardware team. Helping them if they had high level issues with force sensor circuit design. I also managed personal expense reimbursement for the team.

Kemper Siever: Primarily responsible for the design and implementation of all the circuitry for this project. This included designing, testing, and implementing several different force and piezo sensor amplification circuits as well as the power, LED, and switch circuitry. I also managed the parts orders and budgeting for the project. This included the research and selection of the major components such as sensors, MCU, and electronic components. I was also responsible for sensor integration and handle integration.

Wilmot Westriecher: I took care of designing the custom paddle handle and assisting with anything mechanical related. Throughout the project, I researched what the materials we needed sourced for the mechanical section, as well as the microcontroller that we utilized throughout the course of the project. Besides this I learned and took control of all the CAD modelling and 3D printing. Naturally, I ended up setting some of the parameters for the hardware team to keep within. I was also in charge of the majority of the development of the GUI. Additionally, I assisted Oscar with any software issues that he would bring to me, or that we could work on together, this was mostly in terms of features drawn from the IMU.

Abstract

This capstone project, PIKL (Paddle Integrated Kemper Logic), aims to enhance pickleball player performance through real-time data analytics using a smart pickleball paddle. The paddle is equipped with three piezoelectric (piezo) film sensors around its edge to detect hits and impact location. An inertial measurement unit (IMU) sensor tracks the acceleration and rotation of the swings for speed, force, and stroke classification. Data from the piezo sensors and IMU sensor are processed and wirelessly transmitted via Bluetooth Low Energy (BLE) by the microcontroller (MCU). This data is displayed on a computer program that provides real-time feedback per shot and detailed match analysis, including visualizations like stroke frequency and performance metrics over time. PIKL is controlled by an on/off switch and powered by a lithium-ion battery. The printed circuit board (PCB) manages all the circuitry and connections among the electronic components, all housed in a custom 3D-printed handle attached to the existing paddle core. Designed for experienced pickleball players aiming to optimize their training, PIKL provides valuable feedback and data analytics to help improve their pickleball skills.

Background

In recent years, pickleball has surged in popularity, becoming the fastest-growing sport in the United States [1]. Several factors drive this rapid growth, including ease of play, health benefits, and an addictive nature [2]. Underpinning the pickleball craze is the rise of a competitive scene with millions of viewers and year-round tournaments [1]. This influx of serious players eager to improve their game creates a demand for performance-enhancing technologies. For more established sports, data-driven training tools have become essential for athletes [3]. Notable examples include swing speed radars for golf which have been patented since the 1990s [4] or pitch and strike zone tracking technology in baseball [5]. However, as a relatively young sport, pickleball lacks these sophisticated technologies. This technological gap presented an opportunity and motivated the team to develop PIKL, a smart pickleball paddle that provides real-time data analytics on swing speed, stroke classification, impact location, and impact force. The project seeks to fill this void in pickleball and offer experienced players a smart paddle to enhance their game.

In terms of previous projects at the University of Virginia (UVA), PIKL is the first of its kind. While many past projects have utilized sensors to detect force, none have developed a smart paddle. One notable project involved the creation of a robotic pickleball server, which allowed players to train by hitting balls served by the machine. While this project was a pickleball training aid, it did not focus on providing performance or swing metrics, setting the PIKL paddle apart from other projects.

The team is well-equipped to tackle this project, thanks to the diverse academic backgrounds the group has. The team consists of two electrical engineering (EE) majors, one computer engineering (CpE) major, and one double major in CpE and computer science (CS). This mix ensures that the team is capable of handling both the hardware and software aspects of the project. The EE majors bring expertise from courses in applied circuits, electronics, and microelectronics, allowing them to design and implement all the circuitry and sensor integration aspects of the project. The CpE majors bring expertise from courses in artificial intelligence (AI), embedded systems, wireless internet of things (IoT), computer systems and organizations (CSO) 1 and 2, and data structures and algorithms (DSA), allowing them to design and implement the programming and data processing components. With shared coursework, this provides overlap, ensuring that all team members can contribute where needed. This combination of hardware and software experience ensures that both the physical and digital aspects of the PIKL paddle will be designed to meet the needs of pickleball players.

Project Description

Performance Objectives

In pickleball, the sweet spot, located in the center of the paddle, is the ideal area to make contact with the ball, providing maximum shot consistency and control [6]. As such, one of the primary features of the PIKL is pickleball-paddle impact location, which allows users to see how close their shots are to the sweet spot. This data can be reviewed on a shot-by-shot basis or analyzed over entire matches to identify patterns or biases. This feature was present in the proposal and remains unchanged in the final design.

Another important metric in pickleball gameplay is pace or how fast the player hits the ball. This is important to advanced players as faster shots are harder for the opponent to return. Therefore, an important feature of the PIKL is a swing speed metric, which tells the user how fast they hit the ball at the point of impact. This feature was present in the original proposed design and only evolved in its implementation.

Along the same note as swing speed, impact force of the paddle at the moment of impact is important in delivering a powerful shot. Thus, the PIKL provides a classification of the paddle force as either soft, medium, or hard at the time of impact. In the proposal, force was estimated using the force sensitive resistors (FSRs) embedded in the paddle. However, these force sensor readings were found to be inaccurate and were ultimately replaced with piezo sensors. Thus, in the final design the IMU is used to classify impact force through the instantaneous Z-axis acceleration at the impact point.

In pickleball and other racquet sports, there are two types of groundstrokes: forehands and backhands. Different players may favor one over the other, leaving them vulnerable if the ball is hit to their weak stroke side. Therefore, the PIKL provides a groundstroke classification of forehand or backhand for every shot. This enables users to analyze whether they favor certain strokes over time and observe how metrics like swing speed vary with different stroke types. This feature wasn't a part of the original design at all. Instead, we proposed motion dynamics and swing tracing of each shot. After in-depth research of IMUs and motion tracking, it was evident this would be far too complex given the limited time frame. Additionally, we feel the stroke classification feature was a more feasible and useful evolution of the swing tracing feature.

A GUI (Graphical User Interface) program that can wirelessly connect to the PIKL and visualize all of its metrics is important in enabling the player to analyze their game. This feature was present in the proposal and evolved in the final design to include new features such as stroke classification, shot frequency metrics, and match history storage.

Two additional features that remained the same from the original design were the on/off switch and the LED indicator. The on/off switch allows the player to turn on/off the PIKL paddle when not in use while the LED indicator allows the player to visually see if the paddle is on/off. Another feature that was removed in the final design from the original design was the feedback speaker. The final design has the capability to implement this feature, but due to space constraints and user considerations it was removed. Physically, the custom handle is limited with the amount of space, and the paddle is already packed full, so attempting to find a good mounting space for the speaker was not worth it considering the payoff. Furthermore, from an end-user perspective, a feedback tone after every single hit would become annoying and distracting.

In the end, many of the PIKL's original features evolved, as detailed in Table 1 below. However, we feel the final feature set of the PIKL provides a good balance between implementation feasibility in this capstone and utility to pickleball players.

Original Design	Progress	Final Design
Impact Force	Method Modified	Swing Force
Swing Speed	Remains Unchanged	Swing Speed
Impact Detection/Location	Remains Unchanged	Impact Detection/Location
GUI Program	Features Added	GUI Program
Switch/LED	Remains Unchanged	Switch/LED
Speaker	Removed	-
Shot Tracing	Removed	-
-	Added	Stroke Classification

Table 1. Feature Evolution

Technical Overview

To better understand PIKL and its operation, this section provides a comprehensive system overview and explanation of the project. It covers high-level block diagrams seen in Figures 1 and 2, the circuit schematic, the PCB layout, and introduces the key electronic components used.

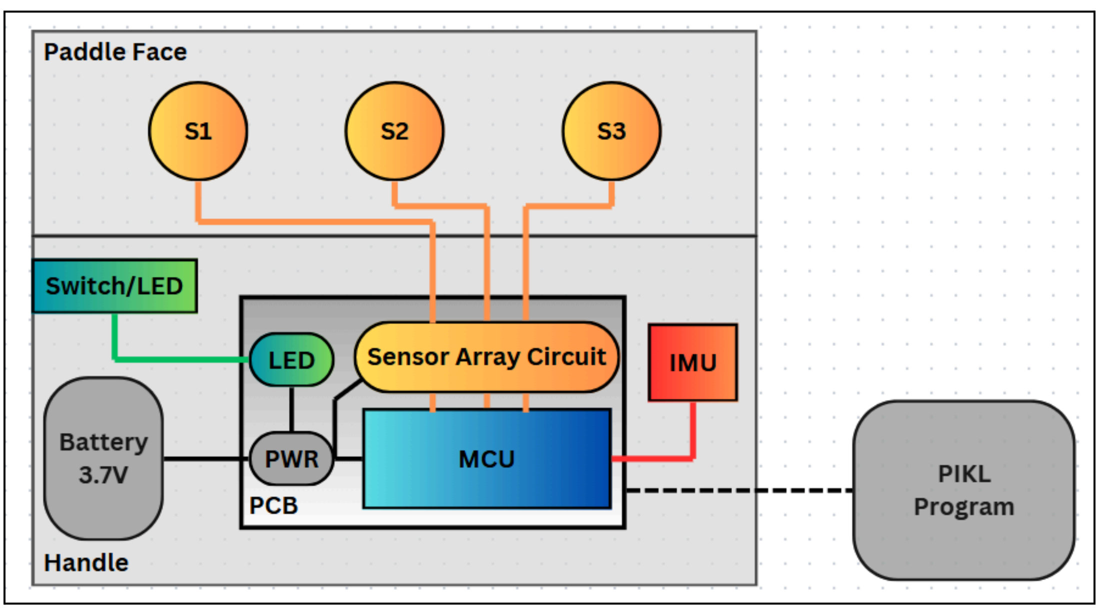

Fig. 1: High-Level PIKL Block Diagram

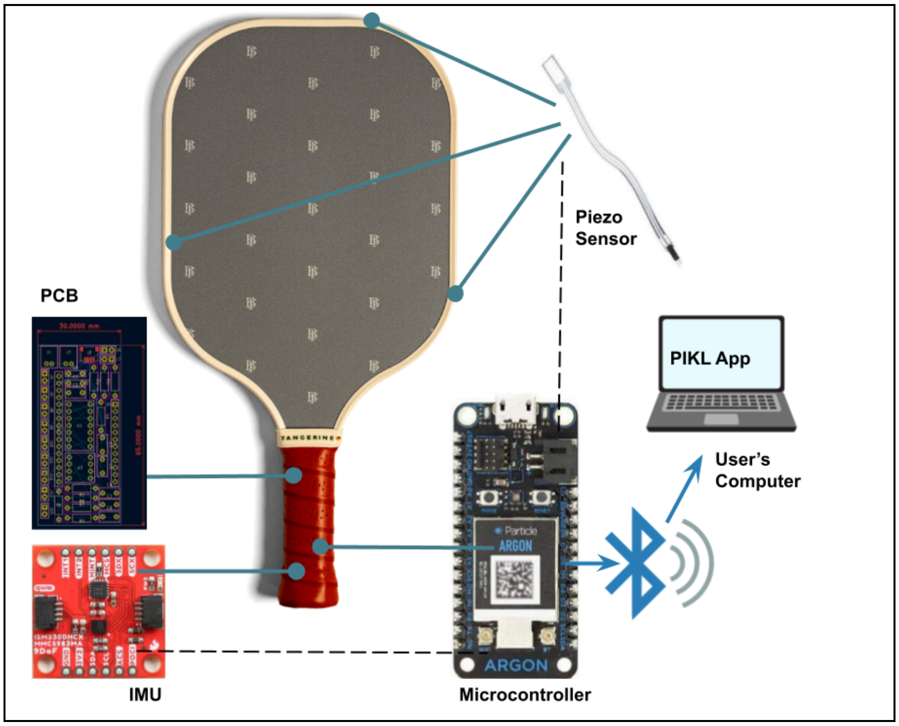

Fig. 2: Component Block Diagram

The PIKL paddle integrates three TE Connectivity piezoelectric film sensors embedded around the paddle edges under the edge guard [7]. These sensors are critical for detecting both the occurrence and location of impacts on the paddle. The working principle is based on the piezoelectric effect, where mechanical stress generates an electric charge [8]. This generated signal needs to be amplified for proper processing, which is achieved through a voltage mode amplifier circuit [9][10]. The detailed design of this circuit is illustrated in Figure 3.

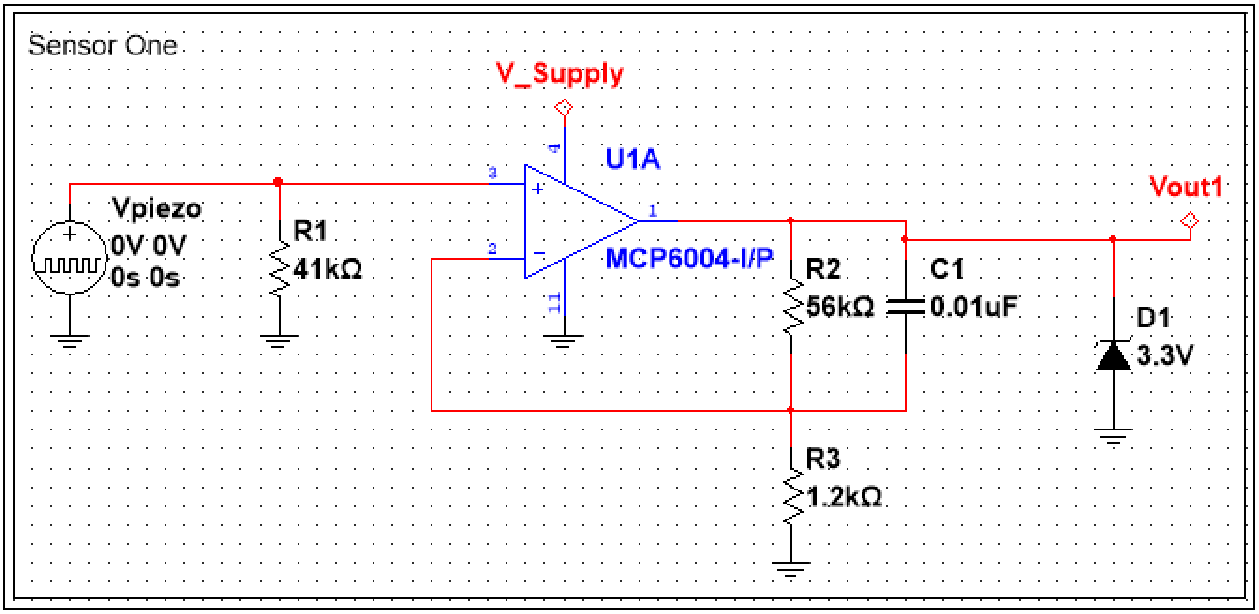

Fig. 3: Piezoelectric Sensor Circuit

The amplifier circuit plays a role in maintaining signal integrity of the sensors. It consists of several key components including resistors, capacitors, and a Zener diode. The resistors are used to set the gain of the amplifier, ensuring the signal from the sensors is amplified for further processing. Capacitors are included to filter out high-frequency noise, thereby stabilizing the signal. The Zener diode is essential for protecting the MCU by limiting the voltage entering the analog-to-digital converter pins (ADC). The full circuit schematic, which also includes the power supply and sensor connections, is depicted in Figures 4 and 5. The PCB layout is strategically designed to integrate the power supply, sensor circuits, and MCU connections, ensuring efficient data processing and reliable wireless communication. This can be seen in Figure 5.

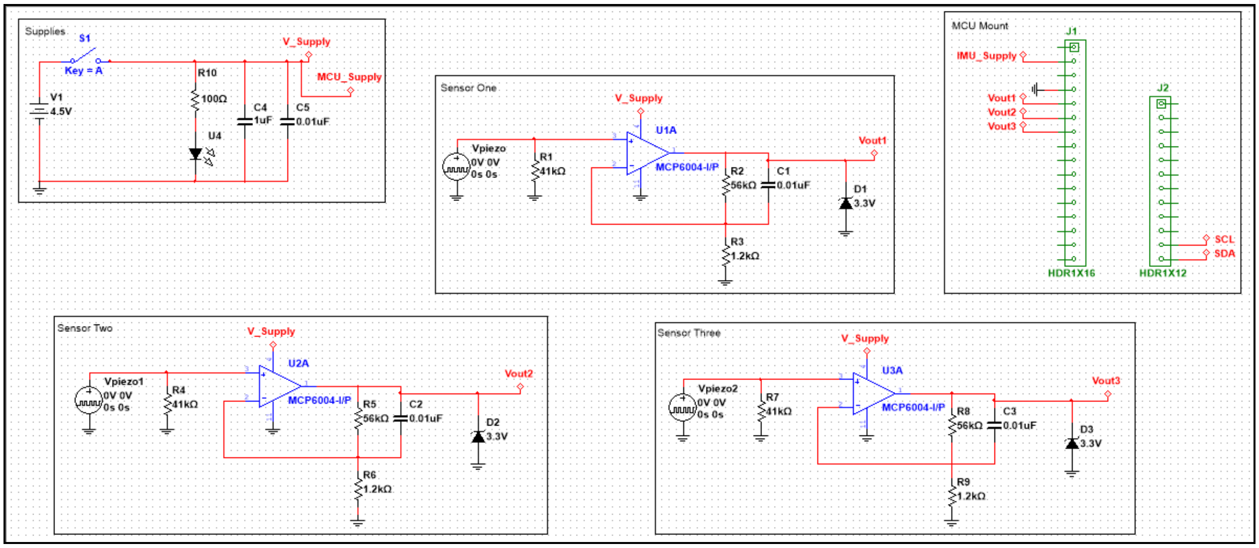

Fig. 4: PIKL Circuit Schematic

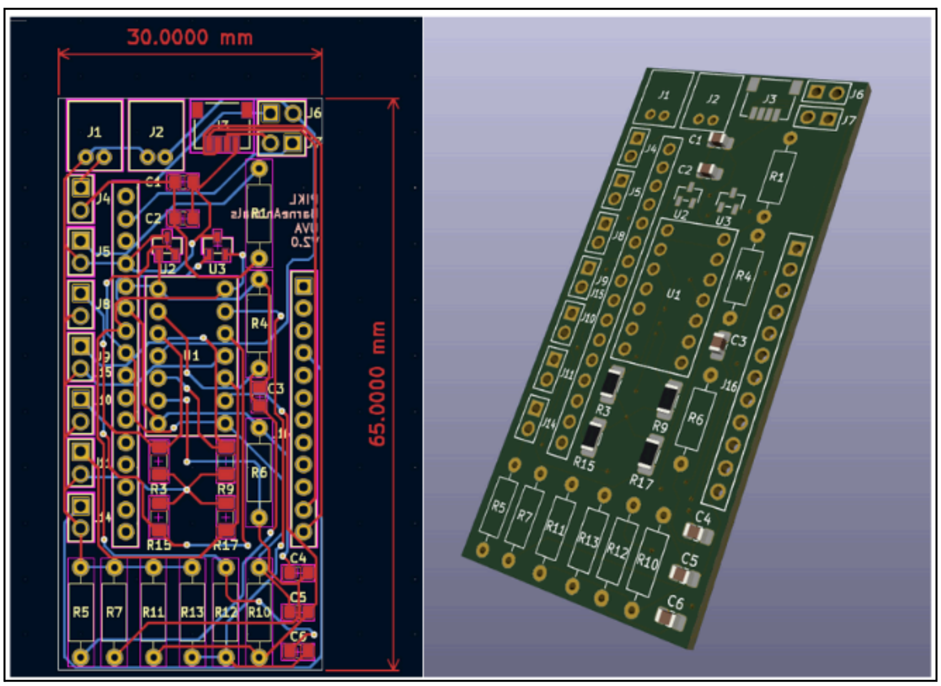

Fig. 5: PCB Layout and Models

The power supply circuit is designed to ensure a stable and reliable operation of the paddle. It includes a 3.7V lithium-ion battery, a switch for on/off, an LED indicator for on/off, and bypass capacitors to minimize noise.

The Particle Argon MCU with an nRF52840 System on Chip (SoC) is mounted on the PCB [11][12]. The MCU processes the data from the piezo sensors and the STMicroelectronics ISM330DHCX IMU [13][14]. When the ADC detects an impact above threshold from a piezo sensor, the shot processing commences. The shot processing uses the accelerometer and gyroscope data collected from the IMU to compute swing speed, impact force, and stroke classification. Once processing is complete, the swing speed, impact force, stroke classification, and piezo ADC impact data are all transmitted wirelessly over BLE to the user device. Here, the BLE data is parsed and processed so that it can be meaningfully visualized in the GUI. A high-level block diagram of the software is shown in Figure 6 below.

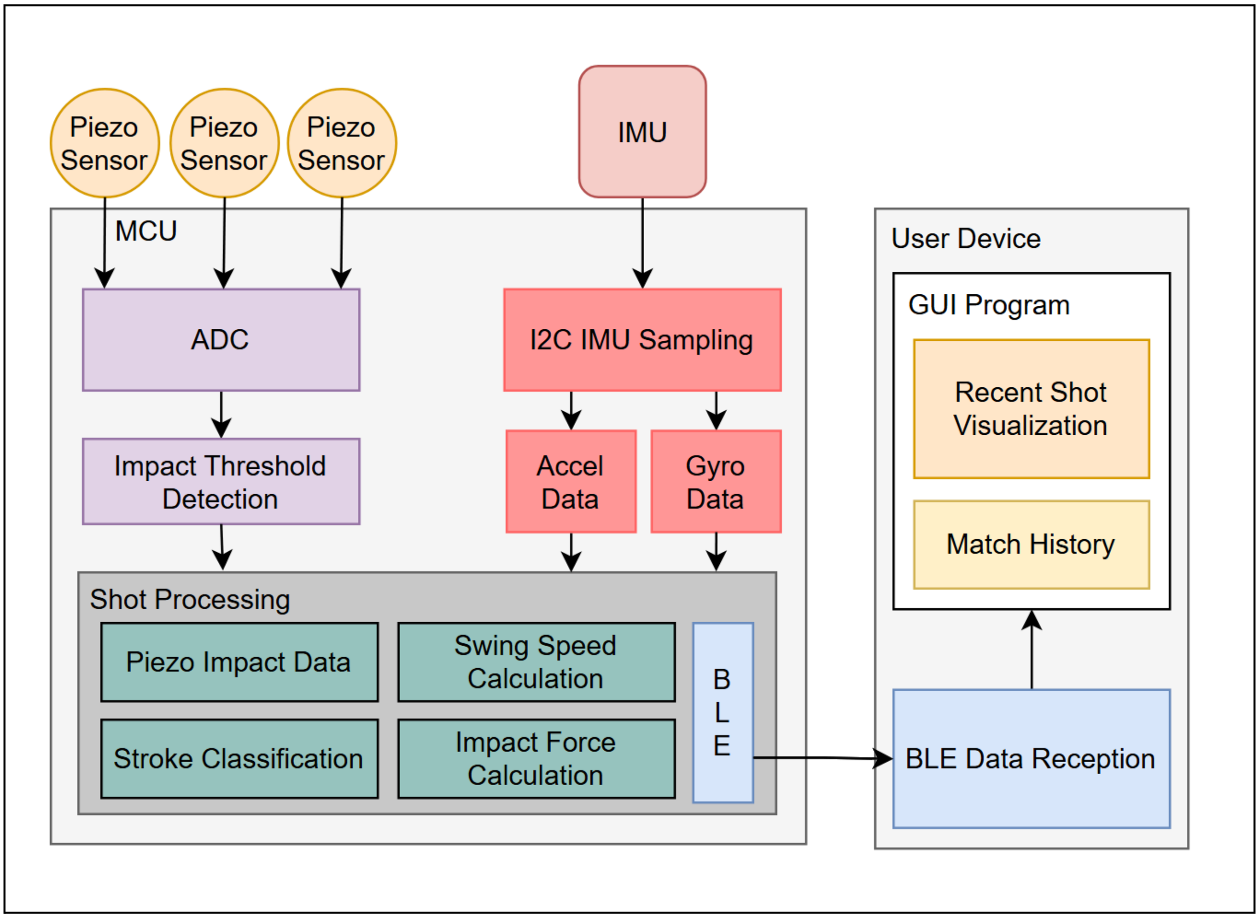

Fig. 6: High-Level Software Block Diagram

Through this overview, we provide a thorough understanding of the PIKL paddle's design and functionality. This section sets the foundation for the hardware and software aspects, which will provide more specific technical details in the section that follows.

Technical Details

Mechanical

In order to add additional components to an existing paddle, we must first understand the paddle's structure. A typical pickleball paddle is divided into multiple sub-parts, this is shown below in Figure 7. The paddle is essentially a honeycomb sandwich panel (HSP) with laminated sheets, an edge guard, and handle support.

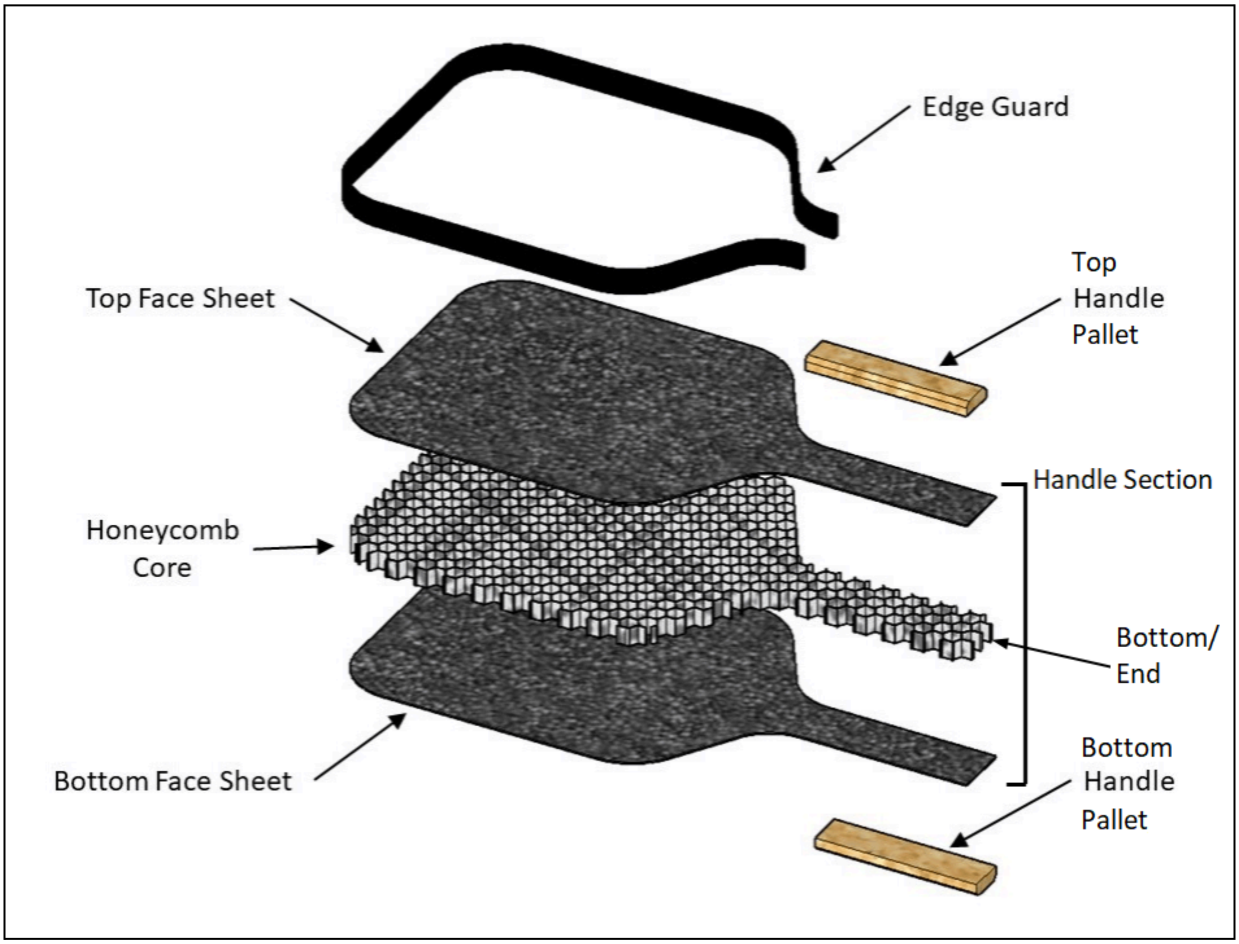

Fig. 7: Paddle Breakdown

Using this breakdown, several design options were explored. These included embedding components, such as force sensors and the IMU, inside the honeycomb core, placing sensors on top of the paddle with protective layers, and implementing sensors on the sides of the honeycomb core beneath the edge guard. The final approach involved placing sensors on the sides, as embedding them inside the paddle face could cause delamination—a phenomenon where the bond between the core and face sheets deteriorates, significantly altering paddle characteristics and creating "dead spots" [15][16]. Additionally, the handle section (handle core) of the paddle was selected to house the other components, as it experiences the least impact force and offers ample space for modifications without compromising the paddle's integrity.

The primary mechanical design for the project centered around a custom handle. This handle needed to house components efficiently while remaining compact and lightweight to preserve the feel of a traditional pickleball paddle. Achieving this required selecting small, lightweight components and materials. The handle was designed to be 3D-printed for cost-effectiveness and material versatility. While initial prototypes used PLA (Polylactic Acid) for its ease of printing and accessibility, the final prototype was printed with Acrylonitrile Styrene Acrylate (ASA), a highly-rated engineering material known for its lightweight properties and durability [17]. After obtaining all the necessary components and selecting the printing material, the first task was to design the handle to accommodate the components shown in Table 2.

Components	Relevant Dimensions (mm)
Particle Argon (MCU) [11]	51 mm x 23 mm x 18 mm (L x W x H)
SparkFun SEN-19895 (IMU) [18]	25 mm x 25 mm x 5 mm (L x W x H)
Custom IMU Holder (PLA) [Measured]	29.10 mm x 29.10 mm x 5.65 mm (L x W x H)
Custom PCB [Measured]	65 mm x 30 mm x 1.6 mm (L x W x H)
Custom PCB w/ Components [Measured]	65 mm x 30 mm x 18 mm (L x W x H)
Lithium-ion Battery [19]	37 mm x 25.5 mm x 5.5 mm (L x W x H)
Wires (standard) [Measured]	1.4 mm thickness of wire, 2.58 mm thickness of connection
M2 Flat Head Screws [20][21]	25 mm long and 5 mm long, 0.4 mm threads
M3 Flat Head Screws [22]	25 mm long, 0.5 mm thread
Heat-set Threaded Inserts [23][24]	4.1 mm and 3.3 mm diameter, 5.9 mm and 4.1 mm installed length
Switch [25]	15 mm x 10.50 mm x 19.20 mm (outer) 13.20 mm x 8.80 mm x 15.50 mm (inner)

Table 2: Components’ Relevant Dimensions and Weights

Considering the listed components in Table 2, achieving a balance between component sizing and functionality was critical. For instance, the initial handle design imposed strict size constraints on the PCB, leaving insufficient space for all necessary circuits. Consequently, the handle design had to be modified, a process that occurred only once or twice throughout the project, resulting in minimal increases in overall size. While the PCB required continuous adjustments during development, components such as the MCU and IMU were less variable since they were selected during initial research. On the other hand, components like the battery and fasteners posed fewer challenges, as even small(er) versions provided sufficient power and mechanical stability for the handle.

The custom handle was designed as a hollow, octagonal structure, consisting of top and bottom halves (caps) fastened together around the existing handle core. This design required all components to fit within the space defined by the caps and the handle core. Initially, a simple approach was considered, placing all components directly into the available space. However, it quickly became clear that this approach would result in a bulky, impractical handle as shown in Figure 8. With guidance from the group's advisor, the design was refined by shaving approximately 2.5 mm off the bottom/end cap. This adjustment preserved the structural integrity of the handle core while creating enough space to accommodate larger components, such as the MCU and PCB, which were stacked together in the expanded section.

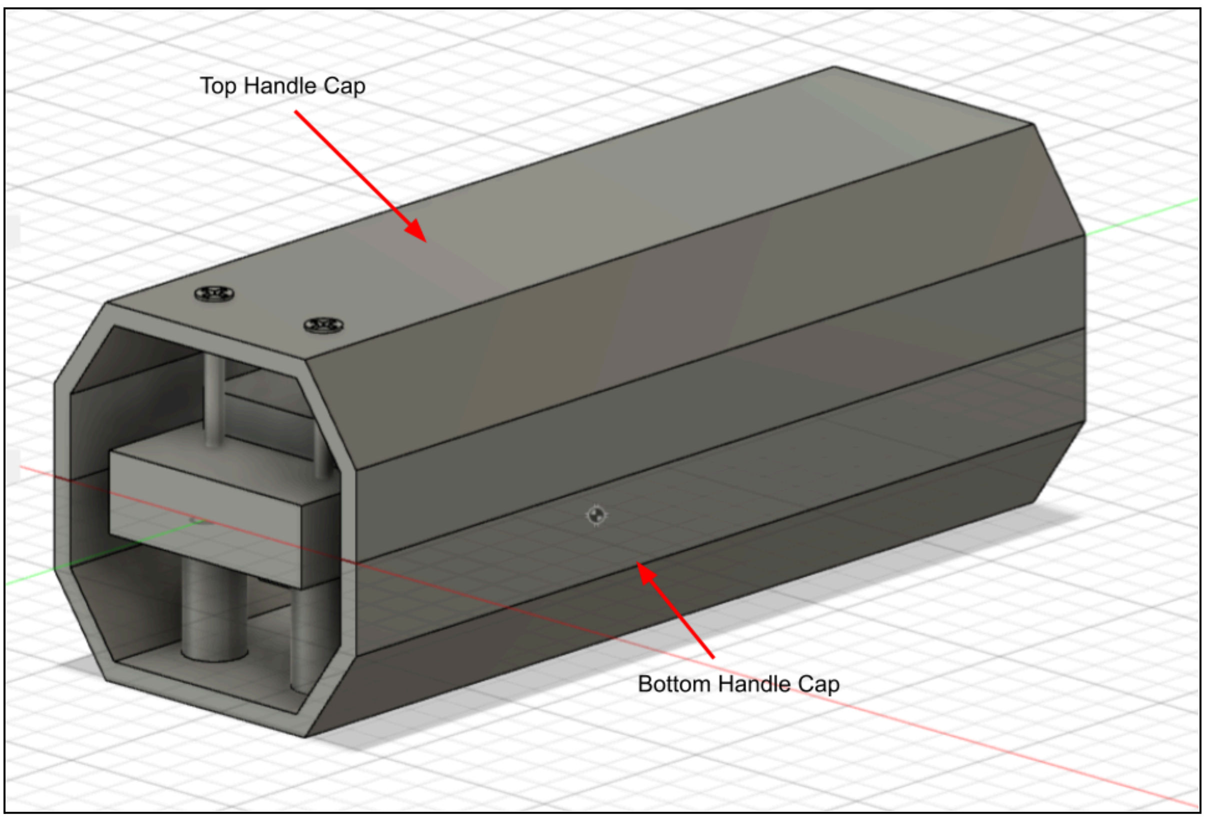

Fig. 8: Initial Design of Handle

This revision marked the beginning of the final design. The paddle handle was lengthened to accommodate the MCU and PCB in the newly expanded section, while the battery and IMU were placed on the intact portion of the handle. The updated design featured a slimmer profile, new slots for the PCB and mounted MCU, a dedicated enclosure for the battery, and columns to support pilot holes and heat-set threaded inserts (HSTIs) for the fasteners. Additionally, countersinks were added at the pilot holes to ensure a flush, secure fit for the screws. Though the countersinks are not visible, all of the other components are in Figure 9.

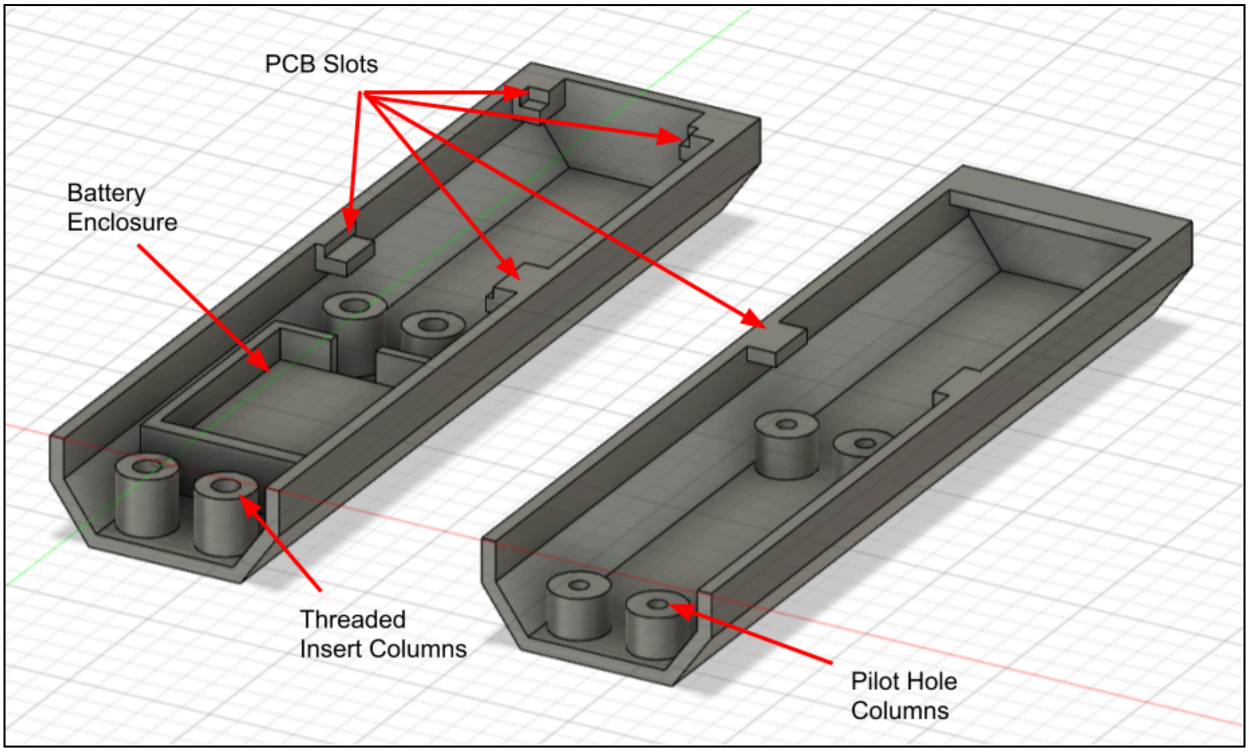

Fig. 9: Second Version of Handle

Unfortunately, the second version of the handle proved to be loose at the back and insufficiently secure. To address this, the next iteration introduced interlocking “teeth” on the sides of the top and bottom handle caps to increase stability throughout the handle and reduce grinding and shear forces when connected. A pair of smaller HSTI columns were added to the rear to provide an additional form of stability at the back. Additional updates included adjustments to the PCB slots to accommodate overlooked corner components on the PCB, a designated space at the back for the switch, and an attempt to create top supports for the slotted PCB. To account for these modifications, the handle was extended by a few millimeters. These changes are illustrated in Figure 10.

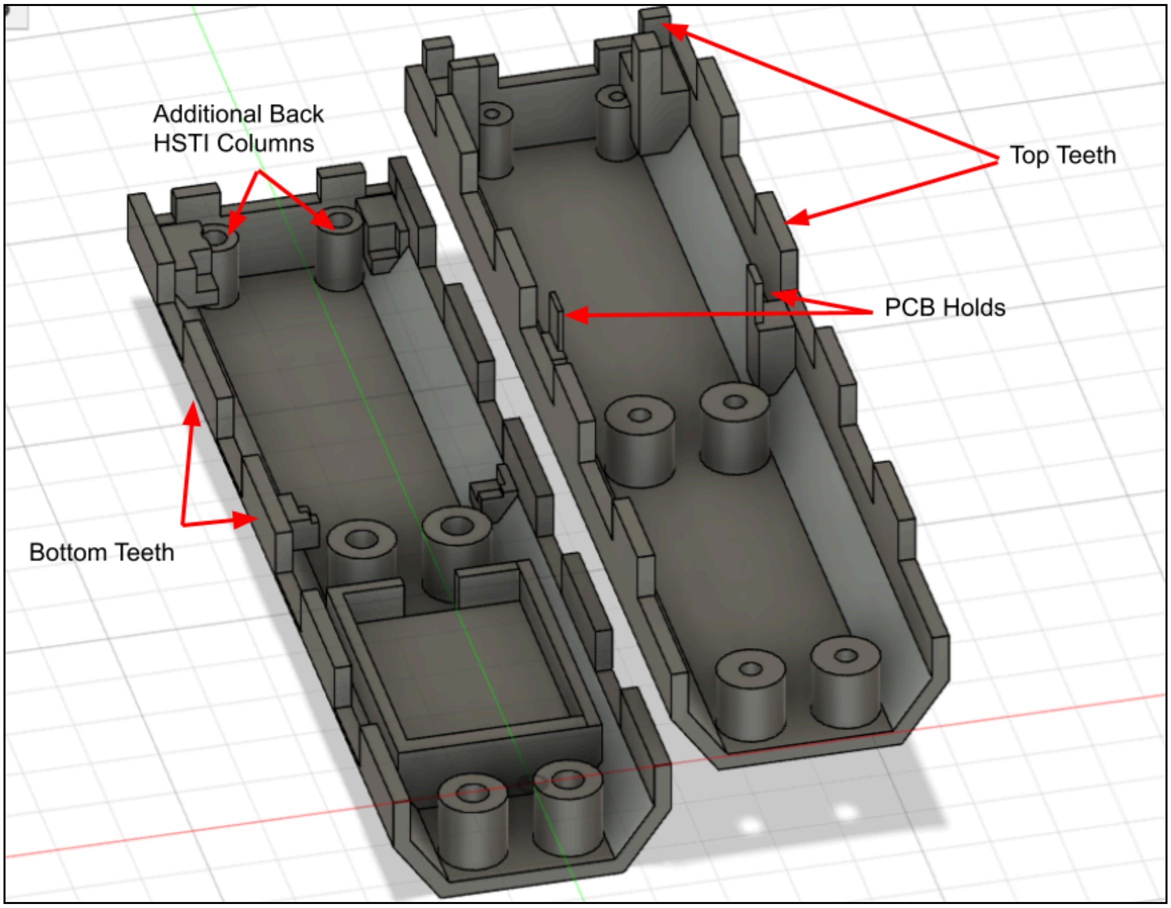

Fig. 10: Third Version of Handle

The penultimate version of the handle included a few smaller refinements. To address an issue where the handle opening pinched wires running from the edge guard sensors into the paddle face, a small jut-out was added at that point. Additional horizontal teeth were incorporated at the top for increased stability, while the PCB holds were removed as they did not secure the PCB any better than the existing slots. The PCB slots were adjusted one final time to accommodate new PCB components and to improve stability by extending the slots' support into the bottom handle cap shell. Finally, a small hole was added near the switch at the back of the handle to house an LED indicator for power status. This final version is illustrated in Figure 11.

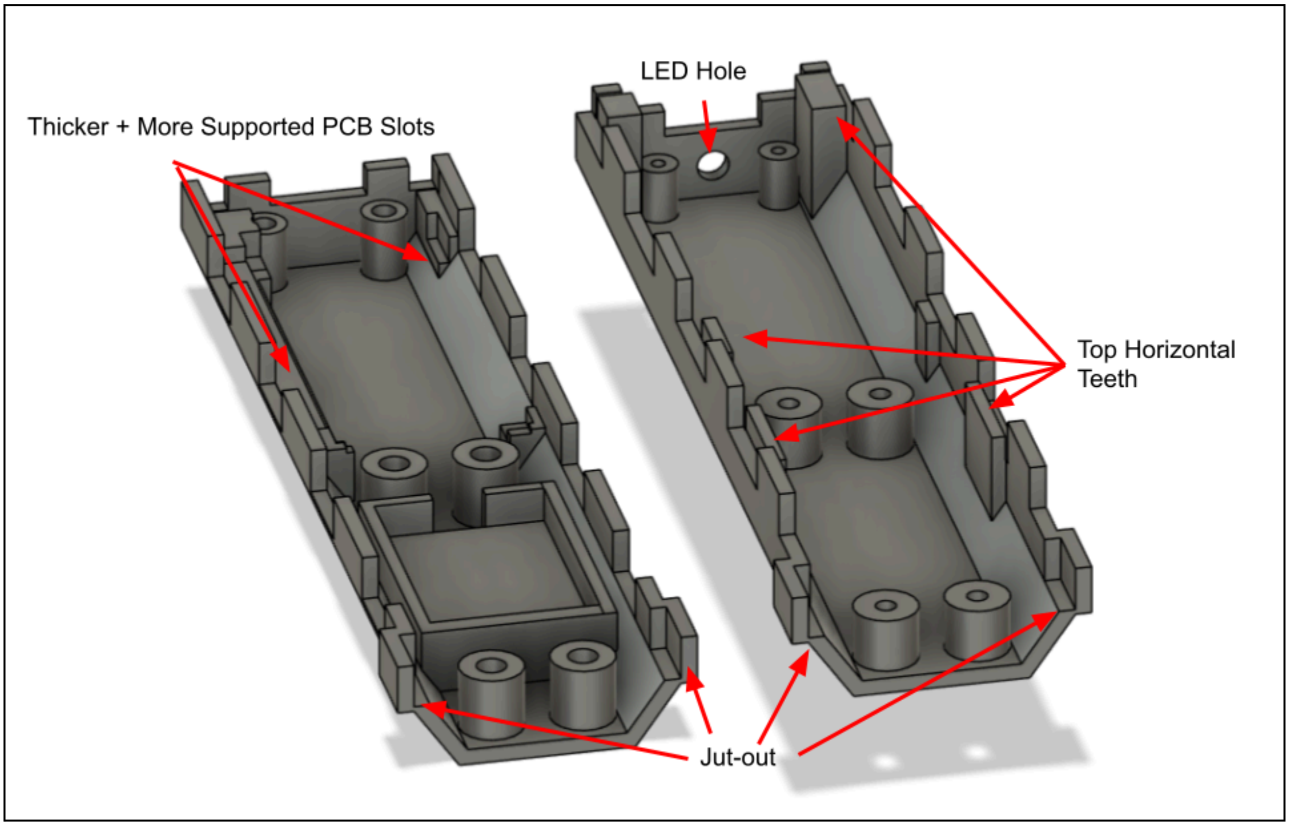

Fig. 11: Final Version of Handle

The final version of the handle resolved all previous issues and provided the most precise fit for both the PCB and the two sandwich caps. While the jut-outs posed a potential overhang issue during 3D printing, supports were added to prevent warping or drooping. Although the handle may not be the most aesthetically pleasing, it fulfills its intended purpose as a functional prototype.

One final note before presenting the paddle's final dimensions: during the design process, the placement of the IMU emerged as a minor challenge. The team determined that the IMU should not simply be placed on top of the plastic face of the honeycomb core (for various reasons, but chief among them is stability and preventing damage to the face/core). To address this, a custom PLA holder for the IMU was designed—a detail previously mentioned in Table 2.

The final handle was printed with ASA and weighed 50.08g and the dimensions were 158 mm in length, 45.89 mm in width at its widest point (including the jut-outs), and 35.40 mm in height. Each jut-out added 2.5 mm to the width on either side. Over the course of development, the handle was extended by a total of 38 mm in length, reduced by 1.11 mm in width, and reduced by 9.6 mm in height from the initial prototype to the final version. While some longitudinal compactness was sacrificed, the final design is significantly more refined in shape and fully meets functional requirements.

Hardware

The hardware design consists of the circuit schematic, specifically the sensor circuits, sensor integration, the power supply, and the PCB design. The full circuit schematic, seen in Figure 4, starts with the battery supply, the switch, the LED circuit, and the bypass capacitors. The battery supply is a 3.7V lithium-ion battery that was chosen based on the needs of the circuit components. This decision is detailed later in the section. The bypass capacitors on the supply provide noise filtering and circuit stabilization. A switch was implemented to allow the user to turn on and off the paddle, while the green LED serves as an indicator for the user to see if the paddle is on or off. The drop resistor for the LED was calculated using the battery supply, the green LED forward voltage, and the green LED current rating [26]. This calculation can be seen below.

$$R_{LED} = \frac{V_{supply} - V_{LED}}{I_{LED}} = \frac{3.7-2}{0.016} = 106\Omega$$

For component simplicity, we chose to set R_{LED} to 100Ω. Next, the sensor amplifier circuit in the PIKL paddle is designed to detect and process signals from the piezoelectric film sensors, which determine an impact and its location. The circuit begins with the piezo sensors (modeled as a pulse voltage signal) that generate a voltage when they experience mechanical stress. The output voltage of the piezo sensors are very small, thus requiring an amplifier circuit. Our design is a voltage mode amplifier circuit, configured in a non-inverting setup, which amplifies the signal's amplitude to be read and processed by the MCU [9][10]. First, the capacitor (C1) acts as a high-frequency filter, filtering noise and stabilizing the circuit, while still allowing our impact signals to pass through. Secondly, the pulldown resistor (R1) ensures that the sensor voltage is properly referenced to ground when no impact occurs, preventing floating voltages. While the specific value of R1 isn't critical, a higher resistance is selected to prevent unnecessary current draw. Thus, R1 is set to 41.2k Ω.

Next, the resistor feedback network (R2 and R3) sets the gain of the amplifier according to the non-inverting gain equation below.

$$Gain = \frac{R2}{R3} + 1$$

In our circuit, the piezo produces a voltage from 0mV to 70mV. Once this signal is passed through the amplifier, the signal is amplified and scaled by our feedback network. Our design has R2 at 56k ohms and R3 at 1.2k ohms. For example, a piezo signal of 50mV, the gain and output voltage can be calculated.

$$Gain = \frac{56k}{1.2k} + 1 = 47.6$$

$$V_{out} = V_{piezo} * Gain = 0.05V * 47.6 = 2.38V$$

At times, the piezo signal does produce a voltage that is larger than the typical range. Due to the op-amp's 3.7V power supply and necessary high gain, the output signal can rail to 3.7V and exceed 3.3V, which is the maximum allowable voltage on the ADC input pins. To address this, a 3.3V Zener diode is placed in parallel with the output signal to limit the output voltage, protecting the MCU's ADC pins from overvoltages.

Lastly, the analysis of the sensor circuit with the capacitor considered is below. V_F is the feedback voltage, which is the signal at the inverting terminal. V_{piezo} is the signal produced by the piezo sensor, which is the signal at the non-inverting terminal. V_{out} is the signal at the output of the circuit, which is the signal sent to the MCU.

$$R_2 || Z_{C1} = \frac{R_2}{1+sC_1R_2}$$

$$V_F = \frac{R_3}{\frac{R_2}{1+sC_1R_2}+R_3} \cdot V_{out}$$

$$V_{piezo} = \frac{R_3(1+sC_1R_2)}{R_2+R_3(1+sC_1R_2)} \cdot V_{out}$$

$$V_{out} = \frac{R_2+R_3(1+sC_1R_2)}{R_3(1+sC_1R_2)} \cdot V_{piezo}$$

$$V_{out} = V_{piezo}(1 + \frac{R_2}{R_3(1+j\omega C_1R_2)})$$

Through this analysis, the design of the full schematic and sensor schematic was finalized. The final designed circuit values can be seen in Table 3 below.

Sensor Circuit	**R1**	**C1**	**R2**	**R3**	**Vdd**
Value	41.2kΩ	0.01uF	56k Ω	1.2k Ω	3.7V

Table 3. Sensor Circuit Designed Values

A major design decision was the battery supply, and the supplies required for all the components. Using the data sheets, the required voltages and estimated current draws were determined. The MCU current draw breakdown can be seen in Table 4 and the total breakdown can be seen in Table 5.

MCU	**BLE Connected[11]**	**nRF52840 Timers[12]**	**nRF52840 ADC[12]**
Current Draw	435uA	418uA	1.24mA

Table 4. MCU Current Breakdown

Supply/Draw	**LED[26]**	**MCU[11][12]**	**IMU[13]**	**Sensor Circuit[27]**	**Total**
Voltage Needed	1.8-2.2V	3.6-4.2V	1.7-3.6V	1.8-6V	n/a
Current Draw	15-20mA	2.1mA	0.7mA	n/a	17.8-22.8mA

Table 5. Supply and Draw for Components

Next, was determining what supply was needed and how to supply it to all the components. Thus the minimum voltage required is 3.6V. With that in mind, a 3.7V lithium-ion battery was selected [19]. This battery is able to supply the LED circuitry, the sensor circuitry, and the MCU. The battery, when fully charged does reach 4.2V, which is still in range of those circuits and the MCU. The LED voltage is dropped by placing a resistor, R_{LED}, in series with the LED. The IMU is the only component that is not supplied by the 3.7V lithium-ion battery. Due to the battery selection, the IMU needed to be supplied with a lower voltage source. To achieve this, the IMU is supplied by the 3V3 output pin of the MCU. With the voltage requirements addressed, the battery capacity was the next decision. Considering that the circuit draws a total current of up to 22.8mA, a battery with a 400mAh capacity was selected. This was determined by doing an energy capacity calculation. The PIKL paddle needs to operate for an entire training session or pickleball match, which is generally an hour. To provide a buffer, we designed for a minimum of two hours of operation. The energy capacity calculation can be seen below.

$$Energy(Wh) = I * V * T$$

The energy is calculated in watt-hours, where I is the current draw in amps, V is the voltage in volts, and T is the time in hours. This example considers our max current draw of 22.8mA, our voltage requirement of 3.7V, and a time of two hours.

$$Energy = 0.0228 * 3.7 * 2 = 0.17\text{Wh}$$

From this calculation, a battery with 100mAh capacity could provide 0.37Wh and suffice, but due to part availability, price, size, and desiring additional capacity for longer sessions we choose to go with a 400mAh battery.

$$Energy_{battery} = 3.7V * 0.4Ah = 1.48Wh$$

This provides the PIKL paddle with a long operating life, while also allowing the battery to be recharged. Therefore, the final design uses a 3.7V lithium-ion battery that has a capacity of 400 milliampere-hours (mAh).

With the circuit architecture designed and the final component values confirmed, the PCB was designed and populated. This design can be seen in Figure 5 in the technical overview section. Our PCB design was constrained in terms of its size. The PCB had to be small enough to fit into the custom paddle handle length wise, width wise, and height wise. Due to the limited space, the MCU had to be mounted on top of the PCB. To do this, we placed female sockets down the middle of the PCB for the MCU to mount on. Underneath the MCU, the op-amp and various resistors and capacitors were soldered on. One side of the PCB consists of testing jumpers for our switch, LED, and piezo sensor connections. The top side of the PCB consists of three connectors. The battery plugs into one of the connectors, while another identical connector provides a jumper connection from the PCB to the MCU. This jumper connector provides the MCU with the battery supply. The last connector is a Qwiic connector for the IMU. This connector connects directly to the IMU and provides power, ground, and I2C (Inter-Integrated Circuit) connections for SCL (serial clock line) and SDA (serial data line) with the MCU. All the components were effectively placed to minimize trace lengths and make connections within the paddle easier. The design of the PCB consists of two layers, a top and bottom copper layer. Both layers are used through the use of vias, while running traces on the top vertically and traces on the bottom horizontally. This was done because of the large amount of traces and connections that needed to be made. The design also consists of through-hole components and surface mounted components. Surface mount components, such as the capacitors and resistors, were used because they are much smaller and take up less space on the board. The PCB underwent testing for all the connections to make sure everything was receiving power and sending data appropriately. The testing results can be seen in the testing and verification section. The final designed PCB can be seen in Figure 12 below.

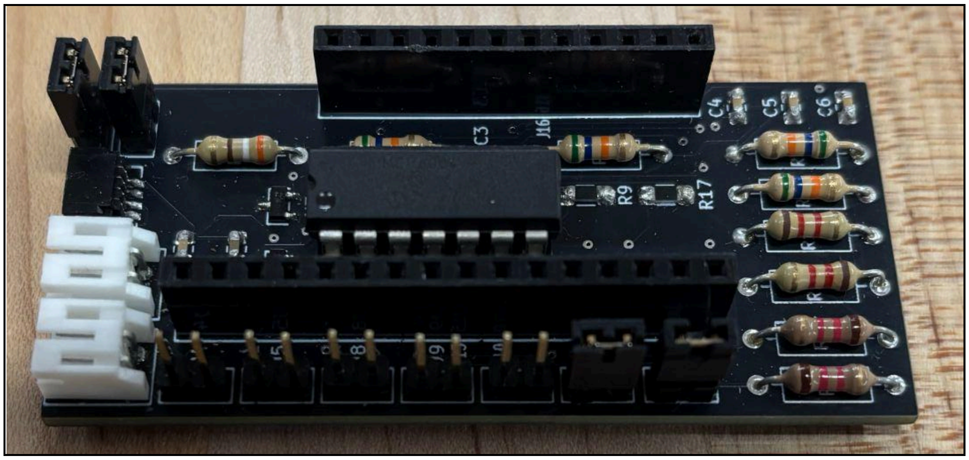

Fig. 12: Soldered PCB Design

One modification from our proposal was the battery supply for the PIKL paddle. Originally, the paddle was going to use either AA or AAA batteries. The concerns with these are size, shape, weight, and capacity. These batteries are generally long, thick, heavy, and cylindrical, which poses problems with space and mounting in the paddle. Additionally, the capacity of AA or AAA batteries are much less than a lithium-ion battery. Therefore, due to these reasons we changed from using AA or AAA batteries to a lithium-ion battery. This battery provides a longer capacity, a slimmer footprint, and the ability to recharge. This modification made the final design more useful for the user as they can use it longer and recharge it.

The major design change that was made was the force sensor switch. In our initial research and testing we looked at both FSRs and piezoelectric sensors. With initial testing we determined that the FSRs were more reliable and stable, versus the piezo sensors which were unstable and sensitive to breathing and vibrations. These tests are discussed in the testing and verification section. With the FSRs selected as our force sensors we began designing the circuitry and performing additional testing. A PCB design was even made and populated for these sensors. However, it became clear that the sensors would not be reliable and consistent. Once we placed them under the paddle face, they were unable to provide accurate force readings. Additionally, the method of placing the sensors under the paddle face caused major structural concerns. With these problems in mind we decided to do some more testing on the piezo sensors, and ultimately chose to start designing the paddle with those sensors. Testing the piezo sensors provided reliable and consistent results allowing us to detect impacts and where they are located. Furthermore, using the piezo sensors enabled us to place the sensors along the edge guard and not create any structural problems. Further details on the testing and verification of both sensors can be seen in the testing and verification section. This design change is an engineering trade off, and presented both pros and cons.

Unlike the FSRs, the piezo sensors are harder to correlate to actual units of force. This new concern is addressed in the software section, with a new force calculation method. Further, due to the piezo sensors being much more sensitive, they can detect impacts better when integrated into the paddle. The change did cost the group both time and additional money. This change also required a new circuit, which in turn required the new PCB design.

Software

A block diagram of the entire software design can be seen above in Figure 6.

ADC

The first embedded sub-component to analyze is the ADC. The outputs from the three piezoelectric amplifier circuits are analog and must be converted with an ADC to be represented in the digital MCU. Thus, each piezo sensor circuit is connected to an ADC input channel on the MCU. From the hardware testing and verification section, the fastest pickleball impact duration was approximately 334 µs or ~2.98 kHz as read by the piezo sensor/circuit. Based on Nyquist's sampling theorem, the ADC must sample at a rate exceeding 5.96 kHz (167 µs intervals) to ensure no impacts are missed and that a good representation of the impact's signal is captured. Furthermore, since the PIKL uses three piezo sensors to estimate impact location, the ADC implementation must support rapid sampling across multiple channels. This requirement presented significant challenges, as many initial ADC implementations failed to meet the necessary speed and multi-channel sampling constraints. The final solution leverages a double-buffered direct memory access (DMA) approach combined with programmable peripheral interconnect (PPI) and a hardware timer. PPI is a hardware feature that enables peripherals to interact directly with each other based on specific events, eliminating the need for CPU intervention [28]. As shown in the list, is the flow of this ADC implementation:

1. Timer-triggered sampling: A hardware timer generates periodic interrupts at the desired sampling frequency.

2. PPI-driven ADC sampling: One PPI channel is configured to trigger an ADC sampling task on each timer compare event. The sampled data is stored in the active buffer.

3. Double Buffering with DMA: When the active buffer is filled with 150 samples, the system automatically switches to a second buffer via another PPI channel. This allows uninterrupted data collection while the CPU processes the filled buffer.

4. CPU Processing: Once a buffer is filled, the processor is notified and processes the stored ADC values. This double-buffered approach ensures continuous data acquisition and minimizes CPU overhead, as the CPU is only involved in processing completed buffers.

During CPU processing of the ADC buffer, each sample from each sensor's channel is checked against the *impact threshold*. From hardware testing below and trial and error, the ideal *impact threshold* is set to 1.5V. If any sample exceeds this threshold, it indicates that the paddle has registered an impact, triggering the start of shot processing. With this setup, up to six channels can be sampled simultaneously at a frequency exceeding 8 kHz, ensuring high-speed, multi-channel data acquisition.

IMU

The next embedded sub-component is the 6-axis IMU sensor, which is used to sense motion and derive critical statistics such as swing speed, impact force classification, and stroke classification. The IMU, specifically the ISM330DHCX chip, includes a 3-axis accelerometer and a 3-axis gyroscope [13].

- The accelerometer, which measures linear acceleration, is configured with a range of ±16 g and an output data rate of 104 Hz.
- The gyroscope, which measures angular velocity, is configured with a range of ±4000 degrees per second and the same output data rate of 104 Hz.

The output data rate of 104 Hz was chosen based on testing that showed an average swing lasts less than a second. With 104 samples per second, per axis, this provides sufficient resolution to capture the motion dynamics of a swing. The configured ranges for the accelerometer and gyroscope were selected based on a study analyzing tennis swings using IMUs, which found that a ±16 g range and a ±2000 degrees-per-second (dps) range were appropriate [29]. However, the study noted that the maximum gyroscope range of ±2000 dps caused clipping during peak rotational speeds [29]. To prevent similar issues in this application, the gyroscope was configured to use its maximum range of ±4000 dps. To accurately interpret the IMU's measurements, it is essential to define its reference frame in the context of the paddle. The IMU is mounted inside the handle of the paddle, with its axes aligned as follows:

- X-axis: Runs through vertically along the paddle handle.
- Y-axis: Runs horizontally along the width of the paddle face.
- Z-axis: Runs perpendicular to the face of the paddle, pointing outward from the hitting surface.

This is further shown for clarity below in Figure 13.

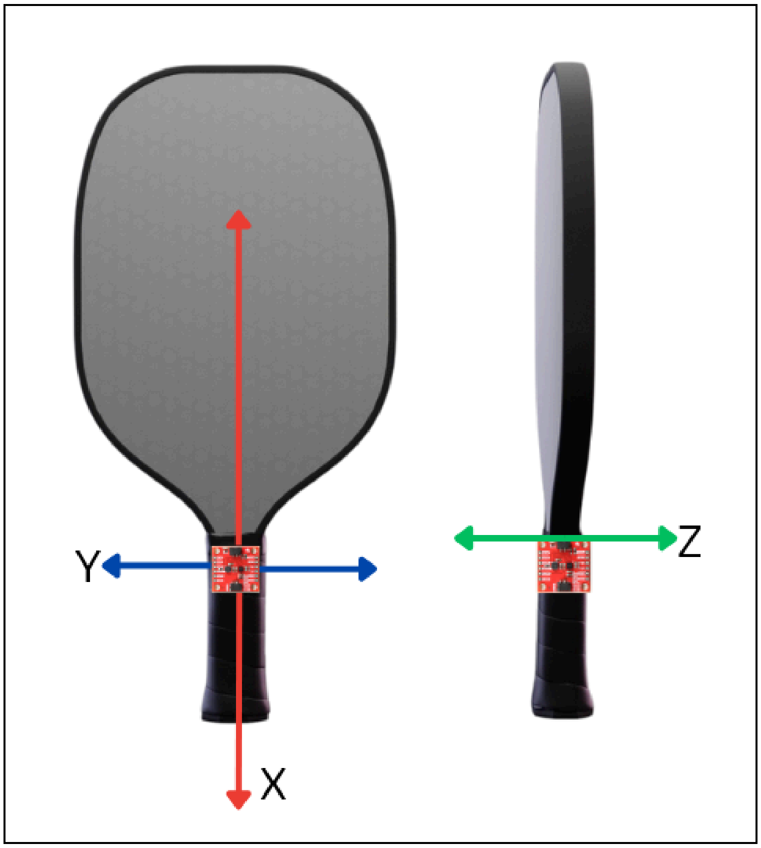

Fig. 13: IMU Reference Frame Axes

The IMU communicates with the MCU via I2C, with Zephyr's built-in sensor API (application programming interface) for the ISM330DHCX. This ease of integration was a significant factor in selecting this IMU sensor. To synchronize data acquisition, a hardware timer generates periodic interrupts at 104 Hz, matching the sensors' output data rate. Initially, IMU samples were fetched over I2C directly within the interrupt handler. However, this approach caused issues because I2C communication is relatively slow and interrupts must be handled as quickly as possible to maintain system responsiveness. To resolve these problems, the timer interrupt handler was modified to submit a work item to the system work queue to read IMU samples. This design offloads the more time-intensive I2C data-fetching process to the main thread, preventing it from blocking BLE transmissions or other critical CPU tasks. The work handler uses the Zephyr sensor API to fetch accelerometer and gyroscope data, storing the results in circular buffers. These circular buffers store the most recent 85 samples, automatically overwriting older data when full. This ensures that, when a shot is detected, the accelerometer and gyroscope buffers contain the most recent swing data, ready for analysis. Another implementation consideration was the size of the IMU data being stored. The sensor values retrieved from the API are 64-bit floating-point numbers. To optimize for a smaller buffer footprint, faster computation, and efficient BLE transmission, the data is converted into a more compact format. Given the accelerometer's range of ±16 g (~160 m/s²), a signed 16-bit integer can represent the data by storing only the integer part and discarding the decimal. Similarly, for the gyroscope's range of ±4000 degrees per second, a signed 16-bit integer is sufficient to store the integer part of the angular velocity. This reduction in data size significantly decreases memory usage and enhances processing and transmission efficiency, without compromising the precision required for accurate analysis of swings and impacts.

Shot Processing

Once the ADC code indicates that an impact has occurred, a work item is submitted to the system work queue to initiate shot processing. The first step in this process is capturing and storing the current IMU sample index as the impact index, representing the exact moment of impact in the accelerometer and gyroscope data buffers. When discussing swing speed, we specifically refer to the velocity vector pointing outward from the paddle face, as this is the direction the ball will travel post-impact. To calculate swing speed, three different methods are used, each with its own trade-offs and considerations.

A critical step in calculating swing speed is determining the rest point of the paddle—the moment when the player's backswing has completed, and the paddle is momentarily at rest, or velocity is 0, before the forward swing begins. To identify this rest point:

1. Start from the impact index in the gyroscope Y-axis buffer (ω_y) and iterate backwards through the buffer.

2. Search for the point where ω_y is close to zero, indicating the paddle has reached its backswing peak and is momentarily stationary.

This algorithm is supported by the graph of the gyroscope Y-axis shown in Figure 14 below. The graph demonstrates how ω_y approaches zero during the backswing's peak, providing a clear marker for the rest point. This rest point is used as the starting condition for velocity calculations, particularly for methods like numerical integration, which assume an initial velocity of zero. By anchoring the velocity calculations to the rest point, the accuracy of swing speed estimation is significantly improved, as the analysis begins from a well-defined and physically meaningful state.

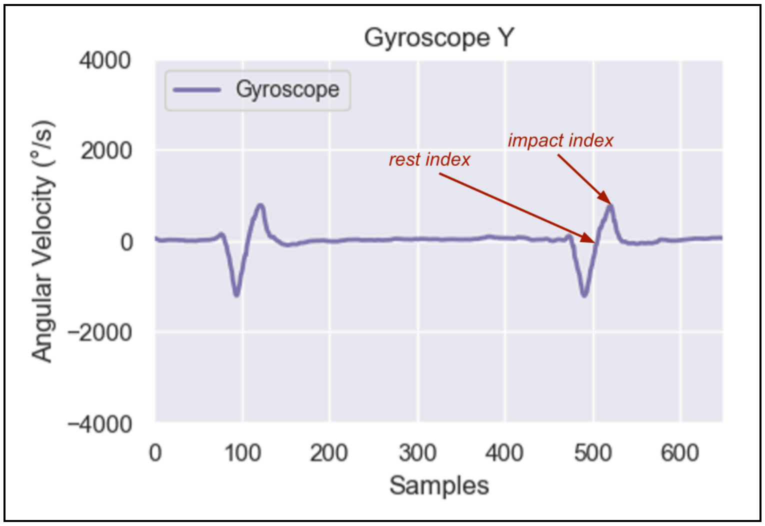

Fig. 14: Plot of ω_y samples showing rest point

1. Numerical Integration of Z-Axis Acceleration:

Description: The Z-axis of the IMU is oriented to point outward from the paddle face. By integrating the Z-axis acceleration (a_z) over time, the outward velocity can be derived using the formula below:

$$v_{swing} = v_{rest} + \int_{t_{rest}}^{t_{impact}} a_z(t)dt \sim \sum_{i=i_{rest}}^{i_{impact}} a_z[i] * t_{sampling}$$

Where $t_{sampling}$ is the sampling interval (~9615 µs).

Trade-offs: Errors in the Z-axis acceleration accumulate over time during integration, reducing accuracy. This method also depends on accurately identifying the rest point discussed earlier. If the rest point is incorrectly determined, the velocity estimation may be inaccurate. Positively, this method returns velocity, allowing it to indicate which side of the paddle face was impacted.

2. Tangential Velocity from Centripetal X-Axis Acceleration

Description: During a swing, a large inward acceleration along the X-axis is induced due to centripetal force. Using the formula for centripetal acceleration, the speed of the swing can be derived:

$$a_c = \frac{v^2}{r} \rightarrow ||v_{swing}|| = \sqrt{a_x[impact] * r}$$

Where a_c is centripetal acceleration, r is the swing radius (~ 0.76 m)

Trade-offs: This method returns the magnitude of velocity (speed), so we cannot indicate the paddle face direction. High centripetal forces during fast swings often cause clipping of the X-axis acceleration, making this method unreliable for especially fast swings (see Figure 15). The method assumes a constant swing radius (estimated at 0.76 m from measuring group's average swing radius), which can vary depending on the player's arm length and swing style.

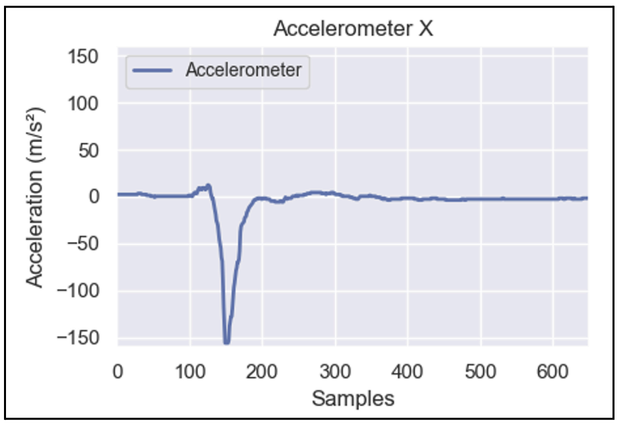

Fig. 15: X-axis acceleration clipping due to centripetal force

3. Peak Y-Axis Angular Velocity

Description: The Y-axis of the IMU runs perpendicular to the paddle handle, making rotation about this axis representative of the paddle's swing. Tangential velocity can be derived using the formula below:

$$v_{swing} = \omega_y * r$$

Where ω_y is the angular velocity about the Y-axis and r is the swing radius.

Trade-offs: Like the centripetal acceleration method, this method relies on an assumed swing radius, which can vary across players. This method returns velocity, allowing it to indicate which side of the paddle face was impacted.

By employing these three methods, swing speed can be analyzed from multiple perspectives, enabling robust estimations under various swing styles and conditions.

To estimate the paddle impact location, the ADC values on each piezo sensor at impact must be transmitted to the GUI. Initially, as soon as an ADC value on any channel above threshold was detected, the ADC values in that moment only were transmitted. This resulted in the paddle impact location estimation in the GUI unreliable. This is because by only transmitting the first samples above threshold, all of the piezo sensors weren't able to climb to their true readings. This essentially made the paddle impact location a function of what piezo sensor rose above threshold first, which was inconsistent. The final solution is to average the next 10 samples from each piezo sensor after impact. This averaging allows a truer representation of the intensity experienced by each piezo sensor.

Another useful metric that is calculated is the impact force. This is calculated by multiplying the Z-axis acceleration at the impact index with the mass of the paddle as shown below:

$$F = ma \rightarrow m_{paddle} * a_z[impact]$$

Where $m_{paddle} = 0.4$ kg

To classify strokes as forehand or backhand, the computed swing velocity is used alongside the Y-axis acceleration at the point of impact. Since the Y-axis is perpendicular to the handle, flipping the paddle from a forehand to backhand position causes the sign of the Y-axis acceleration (a_y) to flip. However, because the paddle has two sides, an additional variable is needed to account for the player rotating the paddle and striking with a different face. Therefore, the sign of the estimated swing velocity (from either method 1 or 3) is used to determine which side of the paddle was used for the hit. The classification logic is shown in Table 6 below.

Signed Swing Speed (Velocity)	**Y-axis Acceleration**	**Classified Stroke**
$v_{swing} > 0$	$a_y > 0$	Backhand
$v_{swing} > 0$	$a_y < 0$	Forehand
$v_{swing} < 0$	$a_y > 0$	Backhand
$v_{swing} < 0$	$a_y < 0$	Forehand

Table 6. Stroke Classification Logic

Bluetooth Connectivity

In order to quickly send data off the PIKL and onto a user's device in real time, the PIKL must implement some sort of wireless protocol. Various options were considered, including Bluetooth Low Energy (BLE), IEEE 802.15.4, and Wi-Fi. The evaluation criteria included range, throughput, power consumption, and complexity [30]. Table 7 below summarizes a simple comparison of these protocols.

Wireless Protocol	**Range**	**Throughput**	**Power Draw**	**Complexity**
Bluetooth Low Energy (BLE)	< 400m	1 Mbps	< 10 mW	Low
IEEE 802.15.4	< 100m	250 kbps	< 100 mW	Medium
Wi-Fi	~100m	> 100 Gbps	> 1 W	High

Table 7. Simple Comparison between Wireless Protocols

In the end, BLE was selected as the wireless protocol due to its low power nature, solid throughput, and familiarity to members on the team. A pickleball court is small and the user's device will be near the paddle resulting in a low required range (< 100 ft). With BLE 5.0, ranges can reach up to 400m as shown in Table 7 above, well above the requirement.

In the PIKL's software architecture, the MCU is configured as a BLE advertiser, allowing the user's device to scan and connect to it. Once connected, the PIKL operates as a BLE peripheral device, acting as a Generic Attribute (GATT) server. This server defines a custom GATT profile with characteristics, or data attributes, for swing speeds (all three methods), averaged impact piezo impact data, impact force, and stroke classification. These characteristics enable the central device (the user's device) to receive notifications from the PIKL. This GATT architecture is shown in Figure 16 below.

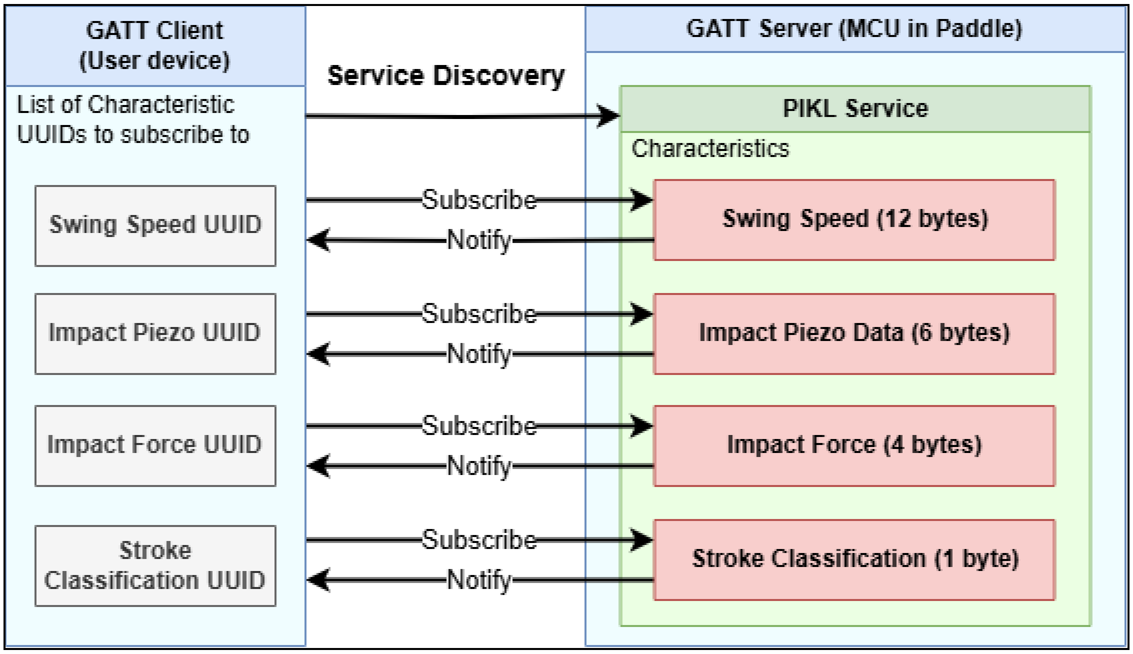

Fig. 16: GATT Architecture Design

BLE has a maximum characteristic size of 512 bytes. From Figure 16, it can be seen that all of the PIKL's BLE characteristics are well below this limit. In the original design, all of the recent accelerometer and gyroscope was transmitted over BLE. This required pushing the BLE implementation to its limits by setting the maximum transmit unit (MTU) to 515 bytes and increasing the number of BLE transmission buffers. While this was an effective solution for transmitting the entirety of the IMU data such as for live plotting, it introduced several issues. For one, transmitting 512 bytes of accelerometer data and then 512 bytes of gyroscope data caused the transmission buffers to fill quickly, sometimes crashing the MCU due to overflow. Additionally, this increased data over the air heavily reduced the operational range of the PIKL. In the final design, almost all shot data processing is done on the MCU (as described above in shot processing). After shot processing is complete and all relevant statistics are calculated, each of the four characteristics, shown in Figure 16, are notified over BLE to the user device. As shown in Figure 16, this means the total size of all characteristics to be transmitted is a much more compact 23 bytes which is swiftly transmitted with BLE's 1 Mbps throughput.

BLE Data Reception

The GUI program must continuously receive BLE notifications from the PIKL's MCU and visualize them meaningfully. To achieve this, the program uses Bleak, a Python BLE GATT client library that interfaces with BLE functionality on the user device [31].

The BLE collection script begins by scanning for BLE advertisements with the device name "PIKL." Once the device is discovered, it sends a connection request. Upon successful connection, the characteristic UUIDs shown in Figure 16 are subscribed to and assigned callback functions. These callbacks are triggered whenever notifications for the corresponding characteristics are received from the PIKL. For instance, when impact force data is received, the callback stores it. After notifications for all characteristics have been received, the script packages the data and sends it over a Transmission Control Protocol (TCP) socket to a specified address and port for visualization. One modification made to this script was to wrap the connection procedure in an infinite while loop with a retry timer. This was added so that if a disconnect occurs, the BLE data collection script will continuously retry connections until the PIKL is back online, making the software more reliable. There is a tradeoff here, as the script will never terminate even if the PIKL is fully turned off, but we feel the robust auto reconnect feature is worth it.

GUI Program

The GUI program serves as the user's interface to visualize the data and insights provided by the PIKL. Written in Python using the tkinter library, the GUI program runs on two threads: one for listening over the TCP socket specified in the BLE collector script and another for managing the GUI interface. In the TCP listening thread, the program continuously listens for incoming data packets. Upon receiving a packet, it extracts the data characteristics. For swing speed, a median filter is applied to the three computed swing speed values to produce a final estimate, which is then displayed on the right side of the interface. The impact force data is classified into three bins—soft, medium, and hard—based on tuned thresholds, with the classification also displayed alongside swing speed. Stroke classification is added as another feature on the right side of the interface. The impact location on the paddle is estimated using the average piezo ADC impact data. This location is mapped onto an image of the paddle displayed in the center of the interface. To achieve this, a coordinate system for the paddle is first established, as shown in Figure 17.

Fig. 17: GUI Paddle Coordinate System

The coordinates of each piezo sensor in the paddle coordinate system are shown in Table 8.

Piezo Sensor	Coordinate (x,y)
Left	(48, 355)
Right	(350, 355)
Top	(231, 583)

Table 8. Sensor Coordinates

To compute the coordinates of the impact location from averaged ADC values from each sensor, the total voltage across all 3 sensors is summed as V_{total} and then a weighted centroid formula is used as shown below:

$$x_{impact} = \frac{V_{left}}{V_{total}} * x_{left} + \frac{V_{right}}{V_{total}} * x_{right} + \frac{V_{top}}{V_{total}} * x_{top}$$

$$y_{impact} = \frac{V_{left}}{V_{total}} * y_{left} + \frac{V_{right}}{V_{total}} * y + \frac{V_{top}}{V_{total}} * y_{top}$$

In addition to displaying data for the most recent shot, the program also stores each shot's data in an SQLite database, enabling detailed match analysis and tracking progression over time. This match history data is accessible via another tab at the top of the interface. Metrics such as stroke frequency, average and max swing speed, max impact force, and an impact location heatmap are available for review. The addition of this persistent data feature was a motivated enhancement aimed at providing users with deeper insights into their performance.

Testing and Verification Plans

Mechanical

The mechanical testing and verification plans revolved around sensor embedding and the handle. The original testing plan was somewhat similar, but instead of testing the paddle face we are testing sensor insertion itself. The reason for this is that there was no exact idea of how the paddle would be instrumented to hold the components. However, now, the test plans are far more specific to the aforementioned sensor embedding and handle testing. The plans were modified from the proposal as follows:

Sensor Insertion Testing

First, the sensors must fit flush against the paddle core and provide expected voltage readings. If these criteria were not met, sensor placement was adjusted accordingly. This process revealed that FSRs were insufficiently sensitive, even when embedded in the paddle core. This was one of the reasons why the design was switched to piezoelectric sensors, which fit neatly beneath the edge guard and provided reliable readings (discussed in the hardware testing section).

Testing was conducted to evaluate the sensors' resistance to damage and accuracy under repeated and higher-impact hits. The sensors demonstrated high durability, withstanding significant impacts on the paddle face without noticeable performance degradation. To validate real-world functionality, the sensors were tested under match-like conditions. The three piezo sensors successfully passed these verification tests, meeting durability and performance requirements.

Handle Testing

The handle was tested to ensure it met design requirements through several steps:

1. Print out: Handle was printed correctly and dimensions were approximately correct.
2. Fit Testing: Verifying that the handle fits securely around the handle core.
3. Component Placement: Ensuring all components fit properly within the handle.
4. Ergonomics: Assessing whether the handle was robust and comfortable to hold.
5. Durability: Determining capability of withstanding repeated swings at various speeds over multiple matches.

The tests were critical as they most often identified the need for modifications and redesigns, detailed in the mechanical subsection of the technical section. Achieving precise dimensions in a compact space posed challenges, making this form of iterative testing essential.

The final handle design utilized ASA material for its robustness, lightweight properties, and suitability for the application. It provided excellent protection for internal components while maintaining durability and ergonomic usability.

Hardware

The hardware testing evolved from the original testing plan laid out in the proposal. Originally, the plan was blocky and unnatural in an engineering sense. The final test plan for the hardware components became more of a natural flowing test plan, that highlights two important tests and verifications. These tests include sensor circuit testing and PCB testing.

Initial Sensor Testing

The initial sensor testing was performed using the following methods:

1. FSR Multimeter Test: The multimeter was connected to the FSR terminals to measure the change in resistance when a finger press occurred or a ball drop occurred.
2. Piezo Multimeter Test: The multimeter was connected to the piezo terminals to measure the change in voltage when a finger press occurred or a ball drop occurred.

The first sensor type tested were FSRs, which produce a change in resistance when a force is applied. Under no force, the sensors have a high resistance on the order of megaohms [32]. Through finger pressing and ball drops, we were able to understand the operation and sensitivity of the FSRs. The FSRs were able to detect larger amounts of force for longer periods of time. The results were consistent and stable throughout these tests.

The second sensor type tested were the piezo sensors, which produce a change in voltage when under mechanical stress. Once again, through finger pressing and ball drops, we were able to understand the operation and sensitivity of the piezo sensors. These sensors are very sensitive, and could detect when we breathed or spoke directly over top of them. They act more like a vibration sensor rather than a force sensor. Additionally, the piezo sensors were unstable, varying in voltage rapidly for any vibrations or pressures.

After those tests, we determined that the piezo sensors were too sensitive, and that the FSRs would be the better option for our application. With that decision, additional research and testing was performed on the FSR sensors.

FSR Testing

The FSR sensors were tested using the following methods:

1. Multimeter Test: Additional multimeter testing was performed on the sensors with finger pressing, ball drops, and weights.
2. Circuit Testing: Tested the sensors with an amplifier circuit and measured the output signals using LabView.

Similar multimeter tests were performed again on the FSRs to verify their operation and functionality. We gained access to a set of weights ranging from 2 to 15 newtons, and placed them directly onto the sensor face. This provided us with a force to resistance ratio to be considered in our circuit testing. The results were reliable and allowed us to design a circuit for the FSRs.

The specific FSRs we purchased and tested included an amplifier circuit provided by the manufacturer. The circuit is an inverting amplifier with a feedback resistor and capacitor. This circuit can be seen in Figure 18.

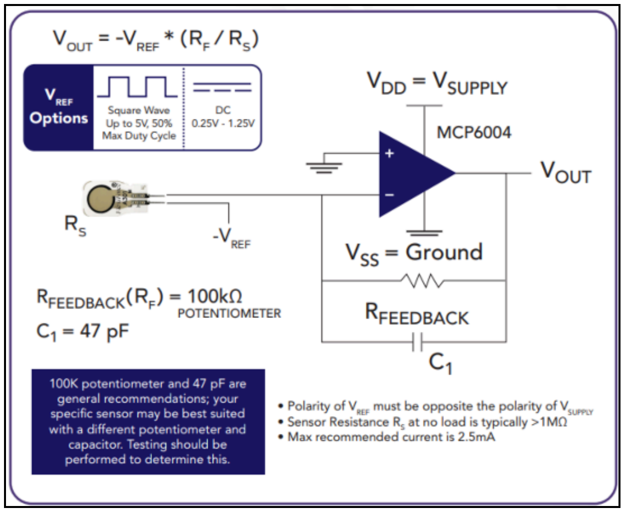

Fig. 18: FSR Sensor Circuit [32]

Next, we built the circuit on a breadboard and performed finger press tests and ball drop tests. Using an oscilloscope and LabView, our results were able to detect impacts and varying impact forces when directly hit or pressed. The sensors were then placed under the paddle face and tested to determine how the results would differ being under a surface. The results were dampened due to being under the paddle face, and they were unable to detect non-direct impacts. To fix this, we attempted to adjust the circuit sensitivity by increasing the sensing range, but to no avail the sensing range was still not wide. Due to the inconsistent results under the paddle face and the small sensing range, we chose to look back into the piezo sensors.

Piezo Testing

The piezo sensors were tested using the following methods:

1. Circuit Testing: Tested the sensors with an amplifier circuit and measured the output signals using LabView.
2. Paddle Swing Testing: Mounted the breadboard circuit to the paddle, connected the AD2 oscilloscope to the circuit and performed small swings.

With the switch to piezo sensors, an entirely new circuit was designed and tested. This circuit and its operation is discussed in the hardware technical details. The amplifier had a resistive feedback network which allowed us to adjust the gain of the circuit. Finger pressing tests and ball drops tests were performed in this circuit for various gain factors, until a reasonable gain was determined and set. These tests provided evidence that the piezo sensors were more sensitive both to impacts and a wider range.

With confidence in our designed amplifier circuit, we mounted the breadboard onto the back of the pickleball paddle. The two oscilloscope channels on the AD2 were attached to the outputs of the sensor circuit on the breadboard. Of the three sensors we have in the paddle, we were only able to view two at a time. A pickleball was thrown at the paddle in different locations of varying force. The primary purpose of these tests were to detect an impact and differentiate between various locations on the paddle. Our tests measured the peak voltage for each sensor and the period of each impact. For the first test, the left and right sensors were connected and measured. Channel 1 is the right sensor (orange), and channel 2 is the left sensor (blue). Figure 19 provides the signals of a middle hit, Figure 20 provides the signals of a left side hit, Figure 21 provides the signals of a right side hit, Figure 22 provides the signals of a top side hit, and Figure 23 provides the signals of a bottom side hit. The results for the peak voltage and impact period can be seen in Table 9,10.

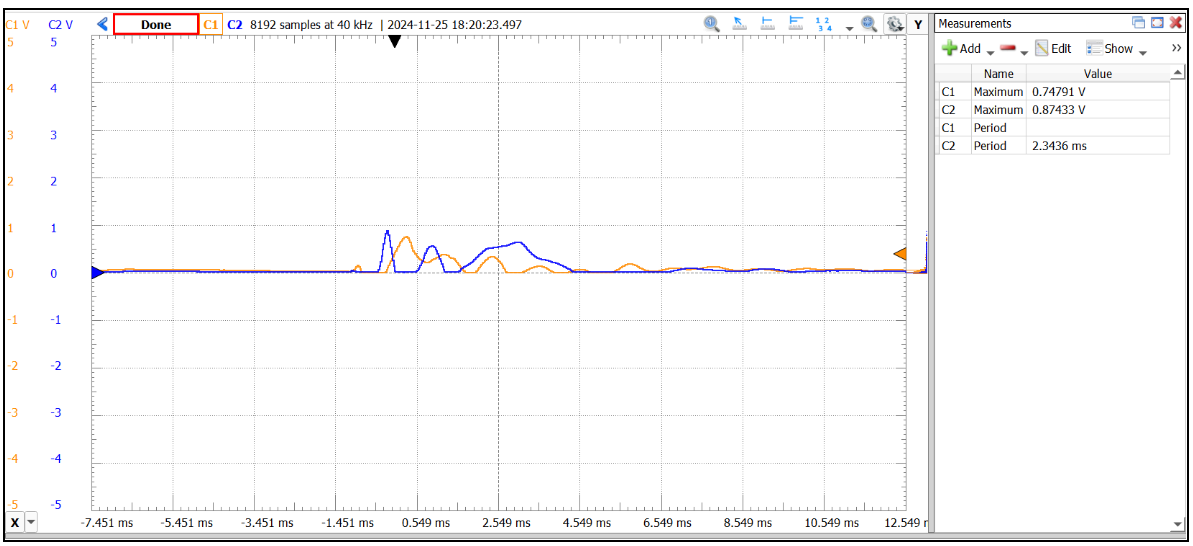

Fig. 19: Middle Hit

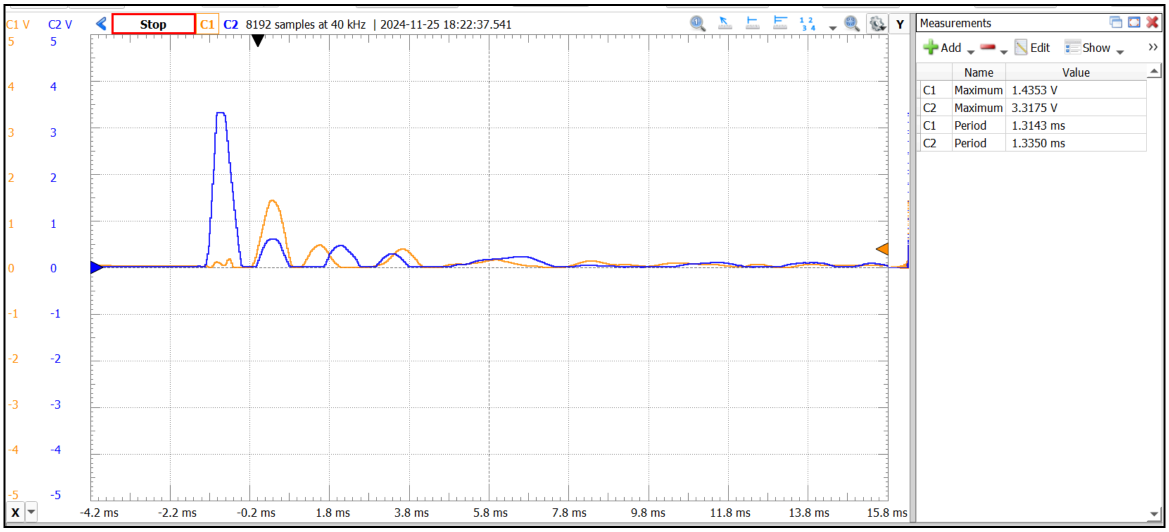

Fig. 20: Left Side Hit

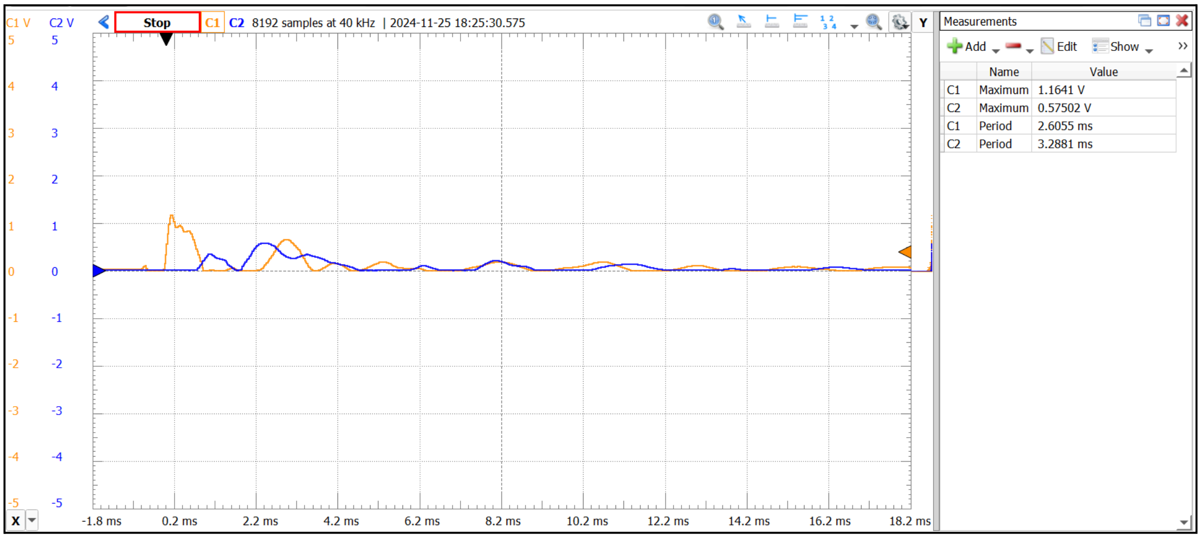

Fig. 21: Right Side Hit

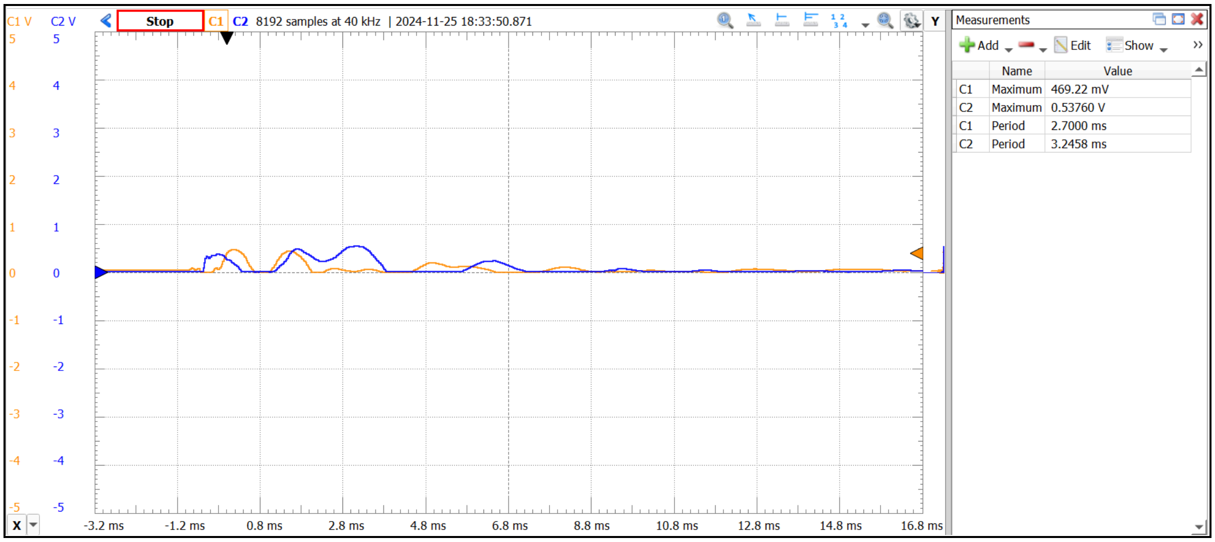

Fig. 22: Bottom Side Hit

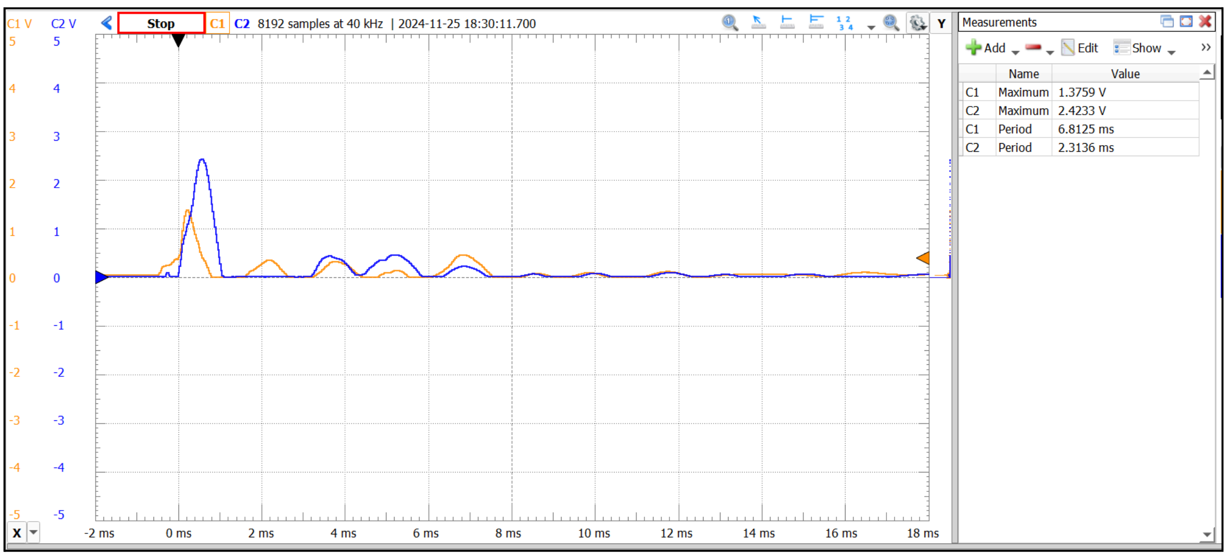

Fig. 23: Top Side Hit

Voltage(V)	Middle Hit	Left Hit	Right Hit	Top Hit	Bottom Hit
C1(right sensor)	0.74	1.43	1.16	1.37	0.47
C2(left sensor)	0.87	3.31	0.57	2.42	0.54

Table 9. Peak Voltage for Right/Left Sensors

Period	Middle Hit	Left Hit	Right Hit	Top Hit	Bottom Hit
C1(right sensor)	1.95ms	1ms	968us	1.32ms	861us
C2(left sensor)	400us	967us	1.05ms	1.69ms	985us

Table 10. Impact Period for Right/Left Sensors

Identical tests were performed with the right sensor and top sensor connected, and with the left sensor and top sensor connected. These results can be seen in Tables 11, 12, 13, and 14.

Voltage(V)	Middle Hit	Left Hit	Right Hit	Top Hit	Bottom Hit
C1(top sensor)	0.52	0.495	0.63	1.52	0.417
C2(left sensor)	1.03	1.90	1.16	0.93	1.33

Table 11. Peak Voltage for Top/Left Sensors

Period	Middle Hit	Left Hit	Right Hit	Top Hit	Bottom Hit
C1(top sensor)	449us	665us	3.42ms	1.33ms	879us
C2(left sensor)	414us	806us	968us	1.28ms	703us

Table 12. Impact Period for Top/Left Sensors

Voltage(V)	Middle Hit	Left Hit	Right Hit	Top Hit	Bottom Hit
C1(top sensor)	0.54	0.476	0.428	1.97	0.413
C2(right sensor)	0.98	1.28	1.77	0.96	0.889

Table 13. Peak Voltage for Top/Right Sensors

Period	Middle Hit	Left Hit	Right Hit	Top Hit	Bottom Hit
C1(top sensor)	801us	334us	1.69ms	966us	650us
C2(right sensor)	598us	360us	897us	1.32ms	809us

Table 14. Impact Period for Top/Right Sensors

From the results we are reliably and consistently able to differentiate between right/left and top/bottom impacts. For example, a left hit should read heavy on the left sensor and much less on the top and right sensors. Some results are not perfectly reflected, primarily top/bottom, but having the third sensor on a channel would have been useful to help us understand how all the sensors are responding. A final test looked at multiple hits in one frame. This was with the left and right sensors being measured. Five hits down the left side, five hits down the middle, and five hits down the right side. This was done to confirm that the impact detection was reliable and consistent. The results of this test can be seen in Figure 24.

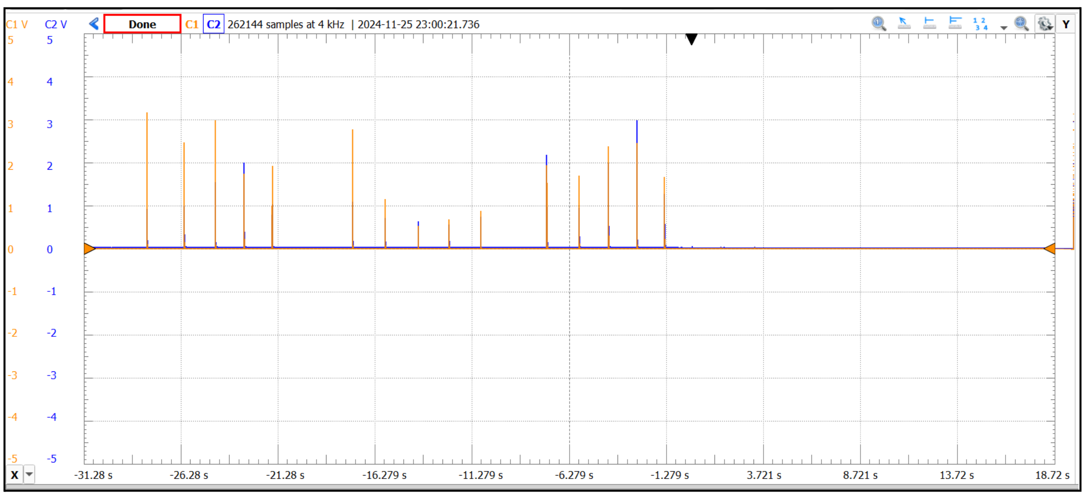

Fig. 24: Multiple Hit Detection

PCB Testing

The PCB testing consisted of the following methods:

1. Visual Inspection: Analyzed the completed PCB for any unsoldered components or incorrect components soldered.
2. Continuity Test: The multimeter was used to test the connections and traces of each pin.
3. Power Supplies: The battery was connected and the multimeter was used to verify that the appropriate pins received power.

4. Sensor/MCU Connection: Connected the sensors, supplies, and MCU to confirm that the MCU connections were able to read the sensor outputs.

With the PCB designed and fully soldered, a visual inspection was performed. This inspection focused on each solder connection to confirm that the connections were all solid and correct. Additionally, the inspection made sure all the components were the correct components and in the correct spots. The solder joints and components all passed the visual inspection test.

A continuity test was then performed on the PCB using a multimeter. This entailed going pin by pin and checking the traces and connections for those corresponding pins. For example, the voltage output pin of one sensor circuit should be traced to the corresponding MCU pin socket. Placing the multimeter on both pins provides a beep if the connection is continuous and traced correctly. All the pins and connections passed this test, which allowed us to perform further testing.

Next, the battery was plugged into the PCB to verify that each component and circuit block received the appropriate supply. One probe of the multimeter was referenced to the ground pin, while the other probe measured the pins that received power. This included the IMU connector, the MCU connector, and the circuit supply for the LED and sensor circuits. Each component and circuit block received the correct power supply.

With the supplies fully functioning, the sensors and MCU were connected and mounted on the PCB. Each of the three sensor circuits were tested using the AD2 at the voltage output pin. A ball was dropped on the sensors to verify that PCB tracing was correct for each sensor circuit. The MCU was then mounted to verify that the ADC pins received the signal from the sensor circuits. Once again, a ball was dropped on each sensor and the output voltage read on the MCU. This verified that the PCB traces to the MCU were correct, and confirmed that the MCU was able to read the signals. Lastly, the system was fully integrated and tested, these testing results can be seen in the system testing section.

Software

Software testing followed a structured, incremental approach, starting with fundamental subsystem functionality before integrating and validating the entire system. This had some parallels with the original test plan, however this test plan has evolved to be more clear in its goal of thoroughly testing each component of the software. Each subsystem—ADC, BLE, IMU, data processing, and GUI—was tested thoroughly to ensure proper operation and performance.

ADC Testing

The ADC in the MCU was tested using a step-by-step approach:

1. Basic Functionality: Dummy data from a digital oscilloscope was used to verify the ADC's ability to accurately convert ground and DC values on multiple channels.
2. Signal Handling: Short bursts of pulse waves were tested to ensure the ADC could detect and convert signals resembling sensor inputs.
3. Integrated Testing: The hardware team simultaneously monitored sensor outputs on an oscilloscope while connecting the MCU through a breadboard circuit to compare ADC readings against the oscilloscope values during quick sensor taps.
4. Final Integration: Once the MCU was mounted on the PCB, the ADC's connection to the three sensors was verified, ensuring accurate data reception and conversion.

As the ADC development progressed, this test plan was followed. DC values were successfully read and converted by the ADC, verifying each channel’s basic functionality. Following Step 2, a 1 ms pulse width wave was generated with Waveforms, emulating a sensor input (Figure 25). The ADC code was able to reliably detect these pulses, verifying sampling could be done fast enough to detect impacts. Lastly, live ADC data from real pickleball impacts on the paddle were plotted as shown in Figure 26. This demonstrated the ADC ability to detect and characterize a paddle impact in real time.

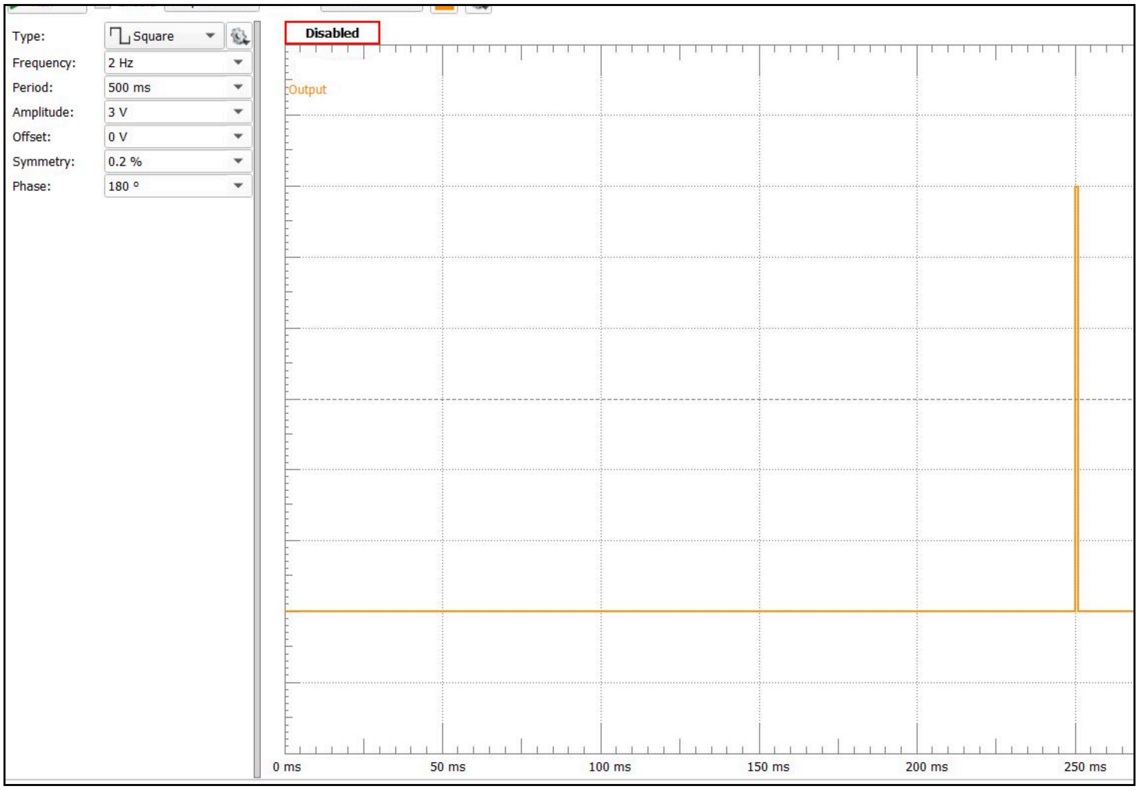

Fig. 25: Input Test Wave Representing Paddle Impact on Sensor

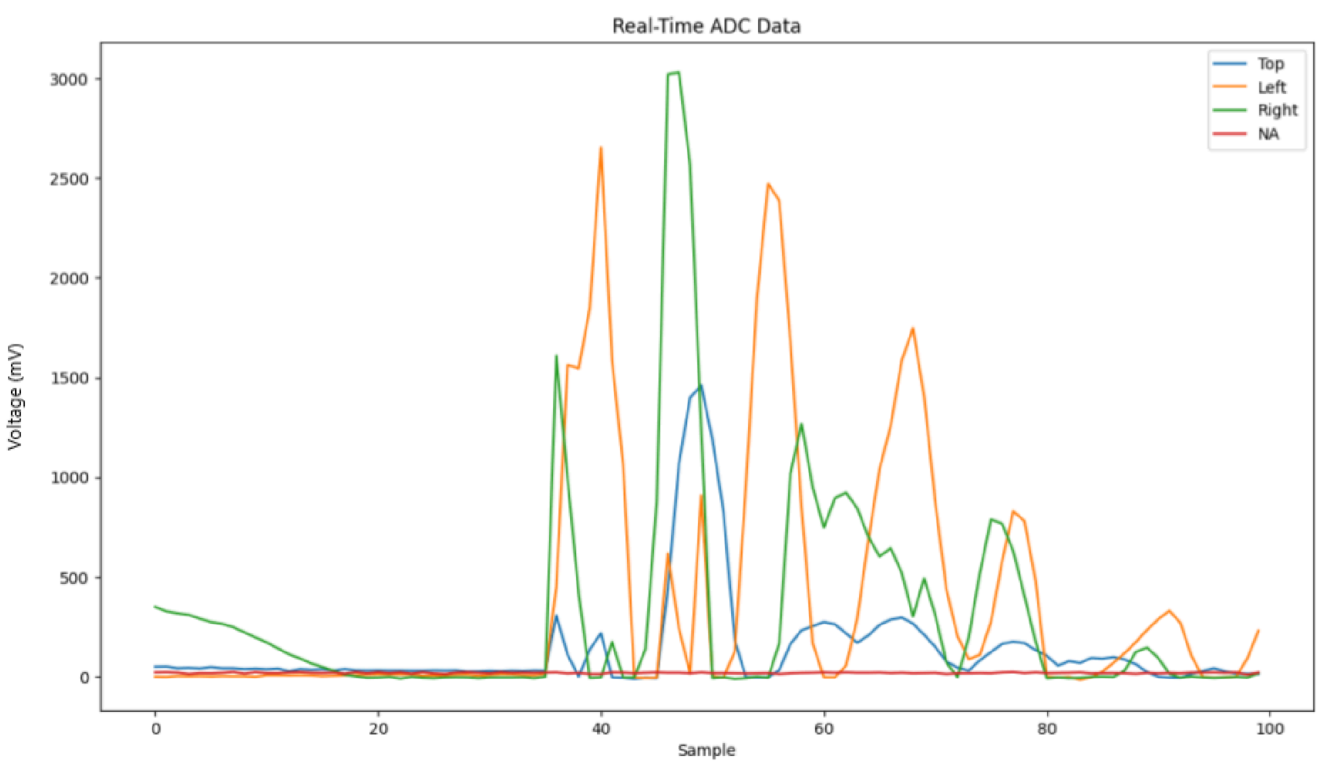

Fig. 26: Live Plot of ADC Data from Impact

BLE Testing

The BLE implementation in the MCU was tested incrementally:

1. Basic Connectivity: A dummy GATT characteristic was created to verify that the MCU could establish a connection and transmit data accurately to a central device (the user's PC).
2. Data Transmission: Real GATT characteristics were created to transmit ADC and IMU data. Logged values on both the MCU and user PC were compared to confirm accurate data transmission.

With the BLE, the first program written was the heart rate sample which is commonly used for BLE connectivity testing. The program was flashed onto the MCU and the MCU was disconnected from the USB of the computer. The MCU should be advertising its signal so that our central (computer) can pick up the advertisement and connect as long as it is searching for its device name, once found the MCU pushes dummy data to the heart rate GATT characteristic. These values are incremented periodically by a set amount, so that let us verify the transmission was reliable. To conclude BLE testing, real sensor data and PIKL calculations were transmitted using the custom GATT architecture defined in the technical details.

IMU Testing

IMU testing was conducted in several stages:

1. Static Accuracy: Initial testing involved logging acceleration data to confirm expected readings. Rotation tests were performed to ensure accurate static readings across all axes.
2. Gyroscope Verification: Yaw, pitch, and roll readings were tested by rotating the IMU. While precise accuracy could not be confirmed, obtaining distinct readings for each axis verified proper operation.
3. Dynamic Testing: Incremental and faster movements were performed to confirm substantial changes in readings.

4. Paddle Integration: The IMU was attached to a paddle, and acceleration and gyroscope data were tested against expectations for various paddle movements. Slow-motion camera analysis compared measured speeds and accelerations to actual motion for further validation.

The first test plan was to start at rest, with gravity acting on a single axis. This allowed us to see if each axis would only display gravity or roughly around gravity values, with the other axis reporting either zero or close to zero. This was followed by some gyroscope testing to see if it was displaying angular velocity on the correct axis. Though the actual speed could not be verified, having a reading on the correct axis confirmed usability. Now that both parts of the IMU were functional, the group started doing faster movements, or movements with more jerk to show acceleration values spike. The spikes would be representative of a perceivable qualitative change meaning that it was at least detecting the movements with some reliability. Finally, the last portion is integration onto a paddle to test how the data is represented through a swing and other motions, the typical forehand is displayed on Fig. 27.

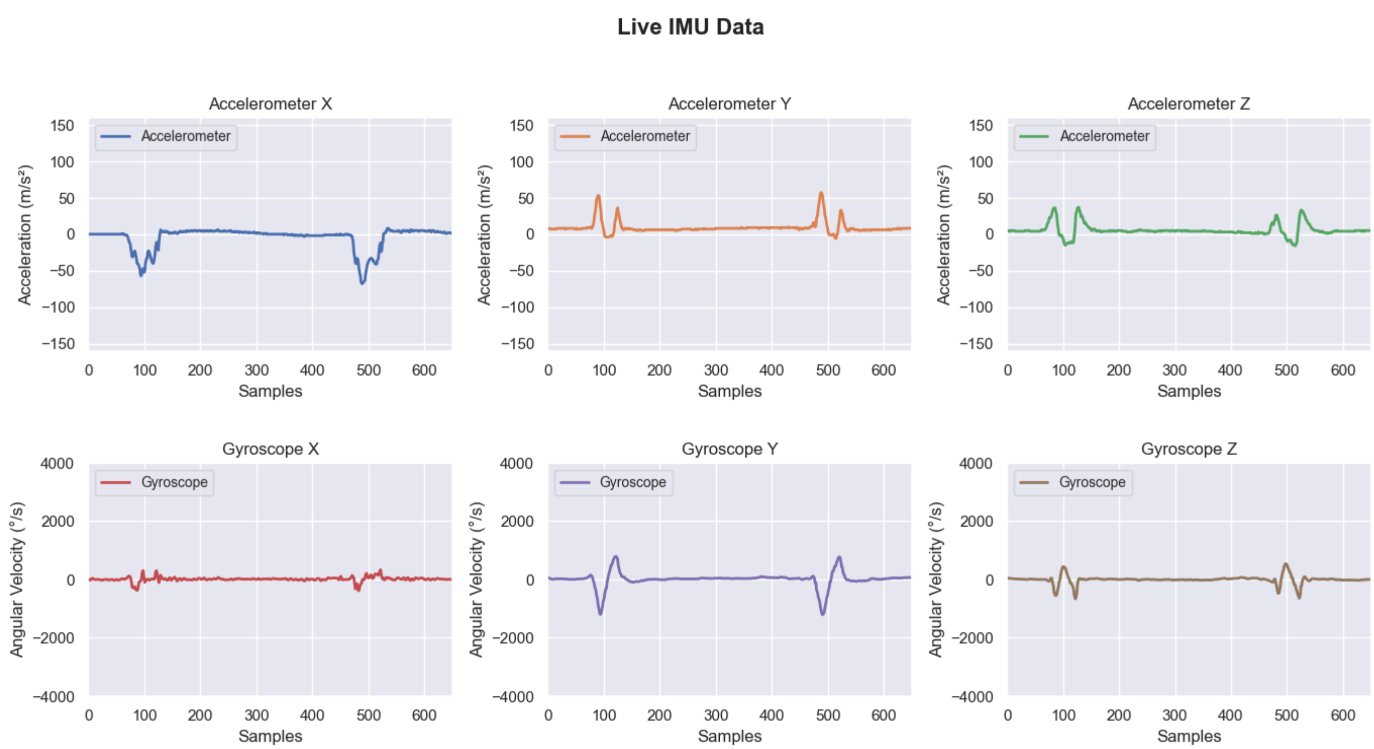

Fig. 27: IMU Data When Attached to Paddle

GUI Testing

The GUI was tested by verifying accurate data display and integration:

1. Data Reception: BLE data reception was confirmed at this stage, ensuring the GUI displayed the correct information that is stored and sent over in BLE arrays.
2. Impact Location: Voltages converted into impact locations were verified using low-tech solutions like marking the pickleball with chalk to observe hit positions on the paddle.
3. Match Data Verification: The database storing match data was cross-referenced with manually collected data to confirm proper storage and retrieval. The GUI display was verified by comparing database values against what was shown on the interface.

Initially, the GUI was tested by generating dummy data. We tested the ability to display fake shot metrics on the interface and estimate a fake paddle impact location. This is shown in Figure 28 below.
Both the true GUI testing and complete testing of shot processing code was completed in the system integration testing below, as truly testing these sub-components requires integration of the whole PIKL.

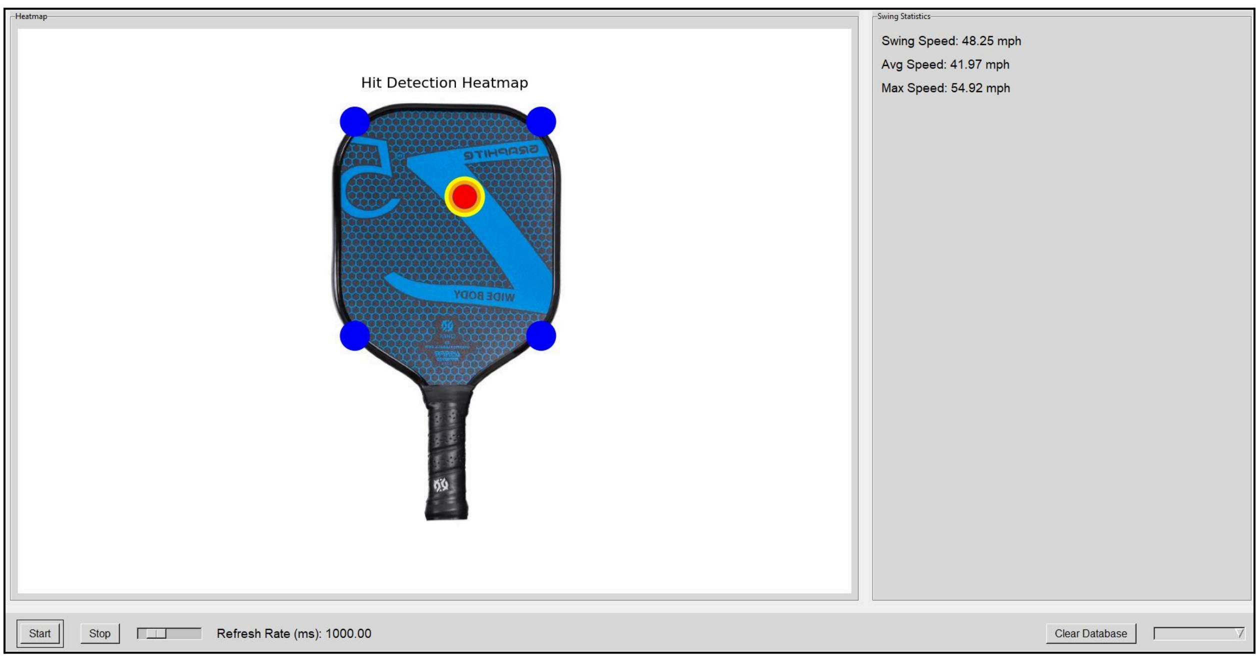

Fig. 28: GUI Displaying Dummy Data

System Integration Testing

In order to fully validate the PIKL's functionality, a comprehensive system integration test was conducted. First, the PIKL paddle is fully assembled in its final form as shown in Figure 29. Then a full system test plan is executed as laid out below:

Fig. 29: Fully assembled PIKL paddle

1. Power-Up: Verify the LED indicator powers on when the switch is flipped.
2. BLE Connection: Ensure the PIKL successfully connects to the GUI program.
3. Gameplay: Simulate gameplay by swinging the paddle and hitting shots, with the GUI displaying shot data in real-time.
4. Match Analysis: Confirm that post-match, the GUI correctly displays match analytics.
5. Accuracy Testing: Evaluate the accuracy of the metrics provided by the PIKL.

The PIKL successfully passed steps 1–4, as shown in Figure 30, which displays a screen capture of the GUI during gameplay. The GUI accurately visualizes impact location, swing speed, impact force, and stroke classification. Additionally, Figure 31 highlights the match analysis tab, showcasing the swing speed trends throughout a match, demonstrating that the PIKL can collect, store, and analyze live match data.

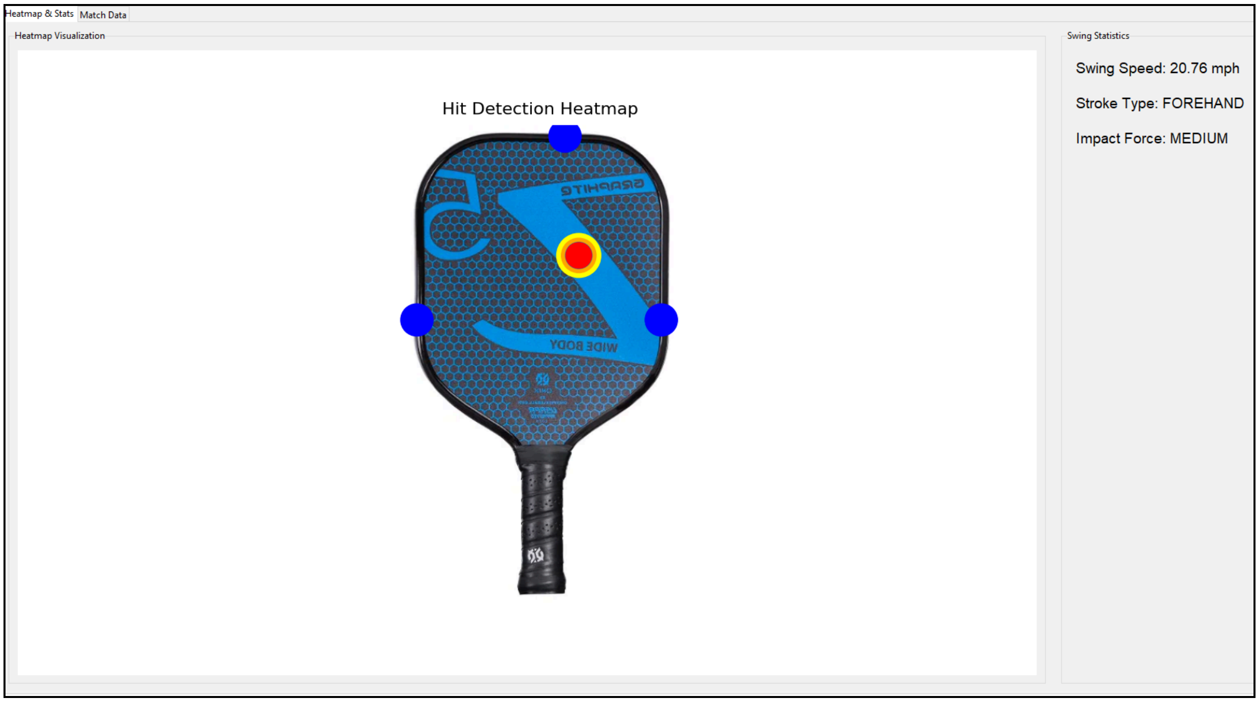

Fig. 30: Screen Capture of GUI showing shot data

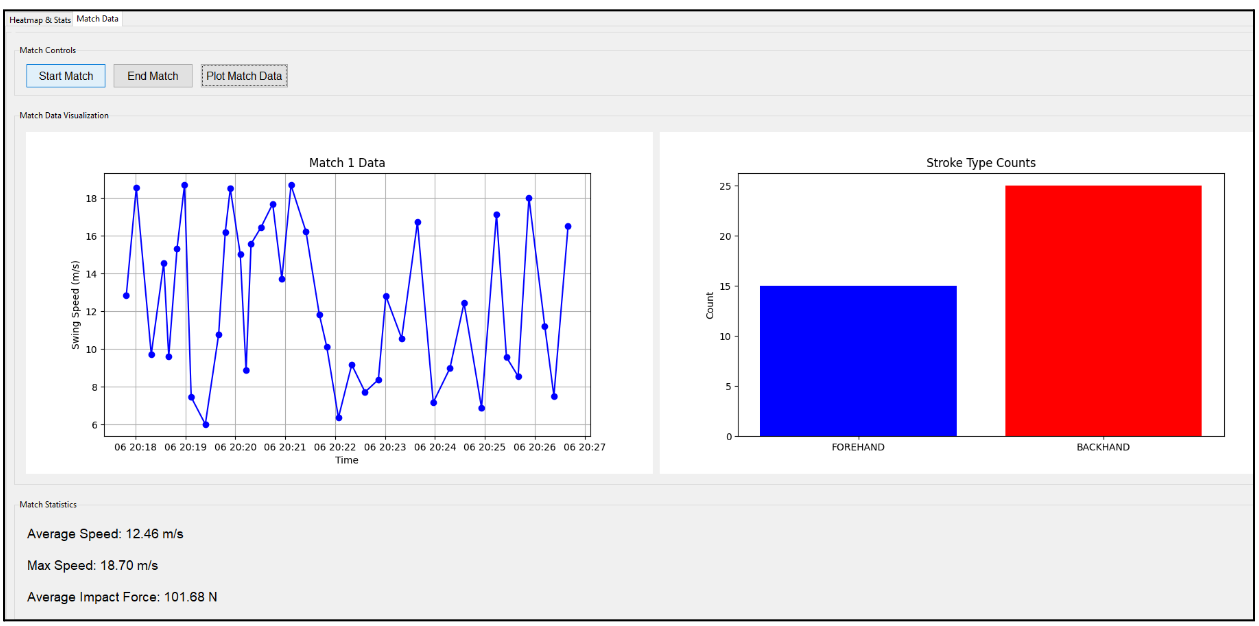

Fig. 31: GUI Match Analysis Testing

From Steps 1-4 in the system testing plan, the final, assembled PIKL is shown to translate pickleball gameplay into visualizable insights and data. However, to ensure full system verification, the quality and accuracy of the insights and data provided by the PIKL must be verified. To do this, several tests are conducted.

While the absence of a radar prevented validation of absolute swing speed with ground truth data, the relative accuracy of the PIKL's swing speed measurements was evaluated. To maintain consistency, a single group member performed controlled swings at three distinct speeds: slow, medium, and fast. Each swing type was repeated 20 times. The results are summarized in Table 15 below.

Swing Speed Type	Average Swing Speed	Standard Deviation
Slow	3.3 m/s	0.71 m/s
Medium	9.7 m/s	1.3 m/s
Fast	17.2 m/s	2.4 m/s

Table 15. Swing Speed Test Results

The results in Table 14 indicate that the PIKL's swing speed metric provides accurate relative measurements. Each swing speed category is distinctly separated, demonstrating the device's ability to differentiate between slow, medium, and fast swings. Additionally, the low standard deviations across measurements highlight consistent performance. While these findings do not confirm absolute swing speed accuracy, they validate the PIKL's capability to deliver reliable relative speed metrics.

To assess the accuracy of the impact location feature, the paddle was divided into five regions: left, right, top, bottom, and center. A team member hit the ball 50 times, with 10 shots targeting each region. Another team member manually recorded the actual impact region during each shot. To ensure accuracy, slow-motion video captured on an iPhone was used to verify the impact location if there was any uncertainty. Simultaneously, the impact location displayed in the GUI was recorded. The testing results are summarized in Table 16 below.

True Impact Location	Correct Location Count	Accuracy
Left	8	80%
Right	7	70%
Top	4	40%

Bottom	6	60%
Center	10	100%

Table 16. Impact Location Testing Results

From the table, it can be seen that overall, the impact location estimate is pretty accurate with a total accuracy of 70% accuracy across all regions. The top region had the worst results with an accuracy of 40%. We think this is due to the top piezo sensor being less sensitive and responsive, despite adjusting its gain. The center region achieved the best results at 100% accuracy.

To evaluate the PIKL's stroke classification, we had a group member hit 60 shots, 30 were forehands and 30 were backhands. The results are tabulated in Table 17 below.

True Stroke	PIKL: Forehand Predicted	PIKL: Backhand Predicted	Accuracy
Forehand	26	4	86.7%
Backhand	10	20	66.7%

Table 17. Stroke Classification Testing Results

The overall accuracy of the PIKL' stroke classification is 76.7%. From the table, when the stroke is forehand, the classifier does better with 86.7% accuracy, while backhand classifications are more inaccurate at 66.7%.

To evaluate the accuracy of the PIKL's impact force classification feature, which categorizes force into three bins—soft, medium, and hard—a controlled experiment was conducted. A team member struck the paddle 60 times, with 20 hits intended to fall into each classification. The hits were calibrated by varying the strength of the strikes: soft strikes were light taps, medium strikes were moderate hits, and hard strikes involved maximum effort.

For each shot, the force classification displayed on the GUI was recorded and compared to the intended classification. The accuracy of the classifications was then analyzed to determine the effectiveness of the system. The results are summarized in Table 18 below.

True Force	PIKL: Soft Predicted	PIKL: Medium Prediated	PIKL: Hard Predicted	Accuracy
Soft	18	2	0	90%
Medium	3	14	3	70%
Hard	0	3	17	85%

Table 18. Impact Force Classification Results

From the results, the PIKL demonstrates high accuracy in classifying soft and hard impacts, with soft achieving 90% accuracy and hard achieving 85% accuracy. Medium strikes had slightly lower accuracy at 70%, with some overlap between soft and medium classifications. This suggests that the boundaries between these categories may need further tuning to minimize misclassification. Overall, the PIKL effectively differentiates impact forces, providing reliable feedback to the user.

Physical Constraints

Design Constraints

Mechanical

From a mechanical standpoint, we faced several manufacturing and resource constraints. One significant challenge was limited access to full-scale machinery and advanced manufacturing tools. While UVA offers excellent makerspaces, they lack certain specialized equipment. For example, we considered using a steel strip to reinforce the paddle core handle and prevent deflection. However, the absence of proper machining tools, such as cobalt drill bits and a drill press, made milling the steel strip difficult. Additionally, access to advanced 3D printers was limited. While simpler 3D printers were available, they could not handle advanced engineering filaments like ASA. As a result, early prototypes had to be made with PLA, which is heavier and more brittle. Eventually, we secured access to a Bambu P1S 3D printer, which allowed us to produce a final ASA handle, but the lack of advanced printing resources earlier in the process posed a significant challenge during prototyping. Another constraint was the lack of information and assistance given by pickleball paddle manufacturers. It made researching paddles an ordeal as all the information is owned by private companies.

Hardware

In terms of hardware, there are several design constraints. Firstly, is the availability and quality of piezo sensors. Due to budget constraints, only cheaper, lesser quality film piezo sensors could be procured. These cheaper sensors had a tendency to break and lose their conductive coatings. Furthermore, suppliers like Newark had low availability meaning only so many could be ordered and more care had to be taken during testing to not break them. If resources were unlimited, more durable and quality piezo film sensors could be ordered in large quantities suitable for testing and production. Another constraint was the PCB layer count. While multilayer PCBs allow for a more compact footprint, they are more expensive and require a more complex design and tracing process. Due to resource limitations, we were restricted to a two-layer PCB. With more time and money, a tighter, more efficient design with three or four layers could have been used. Lastly, is parts shipping and orders. Timing of parts deliveries was always a concern, forcing the group to always think ahead for what was needed. Additionally, some parts were lost during shipping or missed on the order sheet, so there were times when bottlenecks occurred from missing or undelivered parts.

Software

The software aspect of this project was shaped by several constraints. First, regardless of the MCU choice, the embedded software running on the PIKL had to operate on a resource-constrained MCU. This meant that the MCU could not store an entire match's worth of data on board and had to transmit data periodically throughout the match. Additionally, MCUs are limited not only in memory but also in processing power, which requires the software to be written in C due to its lightweight, fast, and memory-efficient nature. Another constraint was limited access to a high-quality external debugger due to budget restrictions. The current setup for flashing and debugging involves using an additional development board connected to the MCU via a ribbon cable, a process that is both tedious and slow. Unfortunately, more performant and user-friendly debuggers were too costly for this Capstone project..

Design Tools

Many tools and applications were used throughout the project for mechanical, hardware, and software aspects. The following sections list the tools and applications in their respective category.

Mechanical

- Autodesk Fusion 360: Fusion was utilized as a CAD software to design any and all 3D models for the paddle. It was used throughout all iterations, so it contains a full history of all edits and versions of all 3D models. Fusion was a new software for the group, so it had to be learned and tested to get a full grasp, and likewise to create better designs.
- Ultimaker Cura/Bambu Studio: Both Cura and Bambu Studio were used for the same purpose of slicing the 3D model from Fusion. Slicing is the process of preparing the model for a specific 3D printer with specific settings. There are recommended settings and profiles made for each printing material as well as settings for each printer, but it requires adjusting to come out with a solid print.
- Ultimaker S3/Bambu P1S: The physical 3D printers were used to print out the sliced models by using a USB or microSD. Both printers have their own settings and adjustable pieces. They are mostly troublesome if the cores/hotends are not extruding the filament properly, which requires troubleshooting skills. Otherwise, the 3D printers were straightforward to learn and utilize. The Ultimaker was for prototyping with PLA and the Bambu for the final handle in ASA.

Hardware

- Multisim: Multisim was used to simulate the various circuit blocks for the project. This included the power supply circuitry, the LED and switch circuitry, the MCU circuitry, and the sensor circuitry. The primary goal of using Multisim was to confirm analytically designed values by performing interactive simulations, transient simulations, and AC simulations.
- LabVIEW: The Virtual Bench program was used for confirmation and testing purposes. Firstly, it was used to verify the PCB design. Secondly, and most importantly, it was used to visualize the output signals from the piezo sensors. From the signal, peak amplitude was measured to estimate hard and soft hits, while the period was measured to estimate the required ADC sampling frequency for the piezo sensors. This program was new to the group, so it took some time to fully understand how to configure the settings and plots to read the appropriate signals.
- KiCad: KiCad was used to design the PCB. This included all the organization, placement, and tracing of the electronic components involved in the circuitry. This tool was relatively new to the group, with one member having some experience. This tool provided the group with the opportunity to improve PCB skills. Due to the size restrictions and amount of components, it was a learning curve to place and trace everything efficiently and effectively with no problems.

Software

- Zephyr RTOS (Real Time Operating System) + ncs (nRF Connect SDK): Zephyr RTOS, which is coupled with ncs for additional Nordic libraries and support, was used as the operating system for the microcontroller. Zephyr was very useful as it has easy to use BLE APIs which were used for creating the PIKL BLE service and transmitting data. Additionally, Zephyr has built-in driver support for the IMU (ISM330DHCX) which makes the IMU code simpler. Zephyr + ncs was familiar to the group member working on the embedded software, but it still had a steep learning curve. There was a lot of learning on how to implement more advanced features like non-blocking, multi-channel ADC sampling or large payload BLE notifications.
- VisualStudio Code (VSCode): VSCode was the primary development environment for all the software. It provided several useful extensions for developing with Zephyr and ncs such as nrfConnectSDK extension, DeviceTree visual editor, and more. VSCode was a familiar tool to group member's working on the software and thus helped speed up the development process.
- GitHub: GitHub was used to store and version control all the software for this project. It managed numerous repositories for proof of concept code, final MCU code, and GUI code. GitHub was super useful in looking through code history and rolling back commits when something broke. It also provided an accessible and easy way for the software development to say in-sync across the team.
- C: All of the embedded software was written in the C programming language. This comes downstream from the Zephyr RTOS used which is entirely in C. The group members working on software had good knowledge of C from prior coursework. While minor C quirks had to be learned along the way, prior experience made using C manageable.
- Python: Python was used to develop end-user applications on the Windows laptop. Python was beneficial for creating quick prototyping GUI scripts to see live plots of ADC data and IMU data. This was useful for verifying BLE and sensor functionality on the MCU. Further, Python was used to develop the final user interface where the user can visualize their pickleball swing analytics. While the group had prior work and classroom experience with Python, there were a lot of areas for learning with the Bleak BLE Python library and tkinter/matplotlib libraries for the GUI being new.

Cost Constraints

Many cost constraints played a role in determining the price of the prototype and a projected production model. For the price of the prototype, the initial research/testing parts, design changes, part availability, and discounted parts all play a role in the prototype price. At the start of the semester, various sensors were ordered and tested before selecting a specific sensor. This incurred additional costs that would not be in a production model. In that situation, the specific sensor would already be selected and research/testing parts would not be necessary. The sensor design change with five weeks left came with additional costs as well. This change added an additional $90 to the spending, on top of the $100 spent on the original sensor type. Through more research, this cost could have been eliminated by choosing the correct sensor to begin with. Once again, in a production model, the sensor would already be selected, tested, and confirmed. Although part availability was not a major concern for the project, this could be a potential concern for a production scale model. In a production model, the parts would be ordered in bulk and this may cause problems in trying to find the right parts, at the right costs, with great quality. Discounted parts and components also play a role for a prototype model and a production model. For piezo sensors, the original price, and price at Digikey and Mouser, were $20 a piece while Newark had them discounted to $7. In this project, this discounted price saved the group money and allowed for the use of these sensors in the PIKL paddle. For a production model, the opportunity for discounts drastically increases when ordering in bulk. This leads to the costs of a production model of PIKL. The bulk ordering and discounts are the primary benefits of a production model. Additionally, the cost of a manufacturer must also be considered. A reliable and high-quality manufacturer needs to be selected to handle the production of a production model. Lastly, promotional and advertising costs can be added to the total. With large quantities of a production model of PIKL, there needs to be enough advertisements and promotions to entice consumers to purchase the paddle. Although there are some similarities between the costs of the prototype and a production model, a production model cost would provide more accurate and reliable budgeting.

Production Considerations

For a production version of PIKL, several key steps need to be addressed. Firstly, rigorous testing is needed to ensure consistent reliability and stability. This includes sensor testing, handle testing, software testing, and paddle integration testing. Refining the code and programming for scalability and performance optimization is also important, to keep up with standard practices. In regards to the code and Bluetooth transmission, security measures must be in place to safeguard against potential cyber vulnerabilities. Additionally, developing detailed documentation, such as datasheets, for users is essential for instruction, maintenance, and future improvements of PIKL. Financial considerations include a budget that covers all aspects of development, testing, and deployment. For development, a manufacturer needs to be selected to make the PIKL paddle. A manufacturer would need to provide a quality product, at low costs and high inventory. This may also include licensing costs and patent costs for the PIKL paddle before it begins full production. For testing, the right parts and designs need to be determined and tested. Parts used in a production model should be inexpensive and high quality, providing the necessary needs for the design. For deployment, the final product cost needs to be considered as well as the selling cost. All the parts and manufacturing costs need to be considered, as well as the selling costs and promotional/advertising costs. To summarize, transitioning PIKL from prototype to production involves professional level testing, coding, and documentation. Financially, a budget for development, manufacturing, and promotional expenses needs to be considered. Through all this, the production model needs to be high-quality and cost-effective, to deliver a successful marketable paddle.

Societal Impact

Players are the primary stakeholders, as PIKL is designed to enhance their performance. First, the PIKL paddle provides insights that improve swing and hit technique. Through the technical features, players can see where they hit on the surface and how fast their swing is. Also, by giving players real-time feedback, PIKL allows players to track their progress and set goals effectively. Secondly, PIKL enhances the player's health and fitness in several ways. Pickleball has a lower risk of injury compared to other sports which can influence players of all ages to play. It motivates players to engage in physical activity, contributing to overall fitness and well-being. Also, pickleball requires a lot of cardio, so it helps improve heart health. PIKL can impact players through skill development and fitness development.

Professional pickleball coaches represent a key secondary stakeholder in the adoption of the PIKL, with the device offering both opportunities and challenges. On one hand, the PIKL can enhance coaching by providing live, data-driven insights such as swing speed, impact force, and impact location. This allows coaches to better understand player performance, deliver more targeted feedback, and improve the efficiency of training sessions, ultimately making them more effective in their roles. On the other hand, the PIKL could pose a threat to coaches by independently providing valuable feedback and automatically recording match data—services that might otherwise require a coach's expertise. As a smart tool, the PIKL paddle introduces a new method for coaching and training, offering a balance between supporting coaches and potentially redefining their role in player development.

Manufacturers in the sports equipment industry can benefit from the development of PIKL in the case it becomes an actual product. As pickleball grows in popularity, the introduction of a smart pickleball paddle can enable industry growth and attract investment. However, the development of PIKL may also lead to competition among similar manufacturers making the same product. Competition can help benefit the economy and sports technology market, but hurt manufacturers making similar products. This may benefit the players and trainers, but challenge the manufacturers.

PIKL impacts the community by promoting engagement in pickleball. The paddle enables players to improve their skills and participate in leagues, tournaments, and casual matches. This increased popularity and involvement fosters social interactions among players and the broader community. Furthermore, PIKL can be beneficial in schools, parks, and community centers, where it can be used to build relationships and connections among players of all ages and skill sets.

The PIKL has potential socio-economic implications. On one hand, its higher cost may make it accessible primarily to wealthier players, potentially creating barriers to entry in professional pickleball and exacerbating a socio-economic divide among top players. Conversely, the PIKL could serve as a cost-effective alternative to a coach, offering less affluent players valuable insights into their game and enabling improvement at a fraction of the cost of hiring an instructor.

Economically, many stakeholders can benefit from the PIKL paddle. PIKL can provide opportunities for pro players and experienced players to receive sponsorships, economically boosting the sport and paddle. Additionally, tournaments and championship settings provide an economic benefit. As PIKL may grow, a sponsored tournament could be held to promote the product and encourage people to get involved in pickleball.

The environmental impacts of PIKL include both current and long-term effects. Currently, there need to be pickleball courts to use the PIKL paddle. The construction of these courts often involves cutting down trees and destroying natural landscapes, potentially disrupting local ecosystems and displacing wildlife. In the long term, the PIKL paddle poses concerns about electronic waste. The piezo sensors, lithium-ion batteries, and electronic circuitry can contribute to this waste if not properly recycled. Electronic waste from lithium-ion batteries can cause fires and release toxic chemicals into the environment. Properly disposing of the PIKL paddle, and considering the location of pickleball courts can help mitigate the environmental impacts.

External Standards

For the PIKL paddle, the PCB adhered to the IPC-2221 standards [33][34]. These standards provide a guideline for reliable and manufacturable PCB designs. It covers standards such as materials, board size and shape, component placement, trace and space widths, vias, and thermal considerations [33][34]. This project primarily focused on the tracing and via standards. The trace widths had to be wide enough to carry the amount of current in the circuit, and the vias had to be appropriately placed to not interfere with any traces or components. By following IPC-2221, the PCB for the PIKL paddle meets the necessary standards.

The design and manufacturing of the pickleball paddle followed the specifications set by USA Pickleball. These standards lay out the requirements needed for a legal paddle, while also noting prohibited aspects of a pickleball paddle. For example, paddles must not exceed a total length of 17 inches as stated in section 2.E of the USA Pickleball Equipment Standards Manual [35]. The custom 3D handle model encapsulates the existing paddle handle while maintaining a total length of 17 inches to abide by the standards. Another consideration was the weight of the paddle. Per the USA Pickleball standards, the weight of the paddle has no restrictions [35]. Although there are no weight restrictions, the group took this into consideration to make the paddle weight feel like a traditional paddle. Considering the engineering design of PIKL, the paddle meets the required standards that our project is impacted by regarding weight and length.

As the nRF52840 BLE module operates as an intentional radiator in the unlicensed frequency band at 2.4 GHz, it is thus regulated under the Federal Communications Commission (FCC) § 15.247 (subpart C) [36]. The FCC states the maximum peak output power for such a device is 0.125 watts or 21 dBm [37]. The project complied with these regulations to ensure the devices do not exceed the peak output power and operate within the designated frequency bands.

Since PIKL is using BLE as a wireless protocol, the wireless aspects of the system adhere to the Bluetooth Low Energy Specification. These standards are laid out in IEEE 802.15.1 [39]. Much of the BLE standards are integrated into the system through the BLE hardware that is provided by Nordic and the BLE Zephyr libraries that are imported. These standards play a significant role in defining the system's constraints and functionality. This in part guides the software architecture and the use of the BLE's GATT profile system of services and characteristics.

Other standards include standardized communication protocols such as I2C and SPI to pass data from the IMU sensor to the MCU. The use of these standardized communications is laid out in the IMU datasheet and adopted in the IMU sensor drivers [13].

Intellectual Property Issues

In evaluating the patentability of PIKL, several existing United States (US) patents relating to pickleball and smart sports equipment were considered. Additionally, two patent-pending products were considered in the analysis of PIKL's patentability.

Patent US20210252356A1 describes a standard pickleball paddle with an inner lattice structure [40]. The patent claims that using a paddle with an inner layer of a specific material composition can improve performance. This is an independent claim as it suggests the concept of an improved pickleball paddle that stands alone. The patent also describes the structural features that make up a pickleball paddle. These claims are dependent as they only provide support and additional details about the independent claim. This patent focuses on the physical construction of a pickleball paddle while PIKL integrates sensor technology into an existing paddle. This sets it apart from this patent and thus there is no conflict. Luckily, a generic pickleball paddle is not patented. So, while PIKL couldn't use this patent's unique lattice structure as its base, it could use other standard pickleball paddles as a foundational base.

Patent US7891231B2 describes an apparatus for monitoring and registering the location and intensity of impact in sports [41]. The apparatus is designed for contact sports like boxing, fencing, and martial arts, using detection systems to monitor impacts. The patent claims that the system can measure and detect the intensity and location of impacts. This is an independent claim as this is the main topic of the patent, and does not rely on any other claims. The patent also describes the specific types of sensors and methods for data processing. These claims are dependent because they build upon the independent claim by adding details about the sensors used in the impact system. PIKL uses piezo sensors to detect impact location and impact intensity, but in the context of pickleball. Our integration in pickleball specifically allows the project to stand alone from this patent.

Patent US11452919B2 describes a Bluetooth enabled ball analyzer and locator for golf [42]. It uses embedded electronics to track the golf ball and provide performance metrics like rotation speed. The patent claims a golf ball with embedded electronics that is trackable and can provide performance metrics. This claim is an independent claim as it introduces the concept of a smart golf ball and functions independently from other claims. The patent describes the specific components and sensors used in the ball to gather and transmit the data. These are dependent claims as they support and provide specifics about the independent claim. While this patent uses embedded sensors to track the ball and measure its rotation speed using Bluetooth, PIKL provides additional features and feedback that do not align with this patent.

Kill Shot Pro is a patent-pending product that claims to help improve pickleball training. It is a system that mounts onto any traditional pickleball paddle and provides feedback on coordination and impact location [43]. The use of an LED array allows the user to practice their hand-eye coordination. Through the use of a speaker, the user gets feedback based on the impact location on the paddle. PIKL shares the goal of determining impact location, but provides the feedback through a live computer program. Although this product is not identical to PIKL, it can help define the patentability of PIKL.

Potenza Smart Pickleball Paddle is also a patent-pending paddle that aims to offer performance analytics to its users [44]. This paddle is almost identical to the PIKL project, but explores more features and different designs. It measures swing speed and impact force, but also analyzes spin detection and rates of the ball. This product also communicates using Bluetooth, but uses a mobile app to view the data. Although this product is similar, it still lacks impact location detection and arranges the force sensors in the paddle face versus the edges in the PIKL paddle. The product provides insights on PIKL and how it can become patentable.

From the analysis, PIKL has unique features that separate it from existing patents and patent-pending products. The analyzed patents still share features that PIKL uses just in different applications. Through the use of piezo sensors around the edges and a central IMU sensor, the performance metrics provide a novel approach to improving pickleball training. Given that similar technologies have been patented in similar contexts, PIKL has the potential to be patentable due to its similar characteristics and distinct characteristics.

Timeline

The original Gantt chart presents the expected timeline at the start of the semester. This provided the team with a set plan, and a one week buffer at the end of the semester for any unexpected setbacks. The legend is shown below in Figure 32, and the original Gantt chart can be seen in Figures 33-37.

Jiseoung Kim (JK)
Oscar Lauth (OL)
Kemper Siever (KS)
Wilmot Westreicher (WW)
ALL
OL + WW
KS + JK

Fig. 32: Gantt Chart Legend

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE	PHASE ONE									
							WEEK 1 (09/09 - 09/13)					WEEK 2 (09/16- 09/20)				
							M	T	W	R	F	M	T	W	R	F
1	**Initial Concept and R&D**															
1.1	Feature Selection	All	9/6/24	9/13/24	7	0%										
1.2	Define requirements and contraints	All	9/6/24	9/18/24	12	0%										
1.3	Research Sensors	All	9/6/24	9/18/24	12	0%										
1.4	System Block Diagram	KS	9/6/24	9/18/24	12	0%										
1.5	IMU Algorithm Research	OL	9/6/24	9/18/24	12	0%										
1.6	Paddle Design Research	WW	9/6/24	9/18/24	12	0%										
1.7	Develop Test Plan	All	9/6/24	9/18/24	12	0%										
1.8	Project Proposal	All	9/16/24	9/20/24	4	1%										

Fig. 33: First Section of Original Gantt Chart

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE	PHASE ONE														
							WEEK 1 (09/09 - 09/13)					WEEK 2 (09/16- 09/20)					WEEK 3 (09/23 - 09/27)				
							M	T	W	R	F	M	T	W	R	F	M	T	W	R	F
1.8	Project Proposal	All	9/16/24	9/20/24	4	1%															
2	**Part Selection**																				
2.1	IMU Sensor Selection	WW, OL	9/11/24	9/18/24	7	0%															
2.2	Force/Pressure Sensor Selection	KS	9/11/24	9/18/24	7	0%															
2.3	Wireless Module Selection	OL	9/11/24	9/18/24	7	0%															
2.4	Microcontroller Selection	WW	9/11/24	9/18/24	7	0%															
2.5	Misc. Electronics Part Selection	JK	9/11/24	9/18/24	7	0%															
2.6	Paddle Selection	WW	9/11/24	9/18/24	7	0%															
2.7	Parts Order	OL	9/18/24	9/18/24	0	0%															

Fig. 34: Second Section of Original Gantt Chart

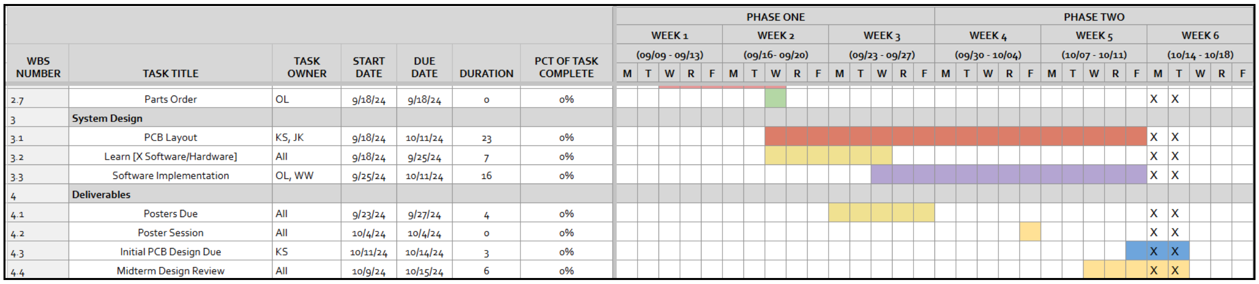

Fig. 35: Third Section of Original Gantt Chart

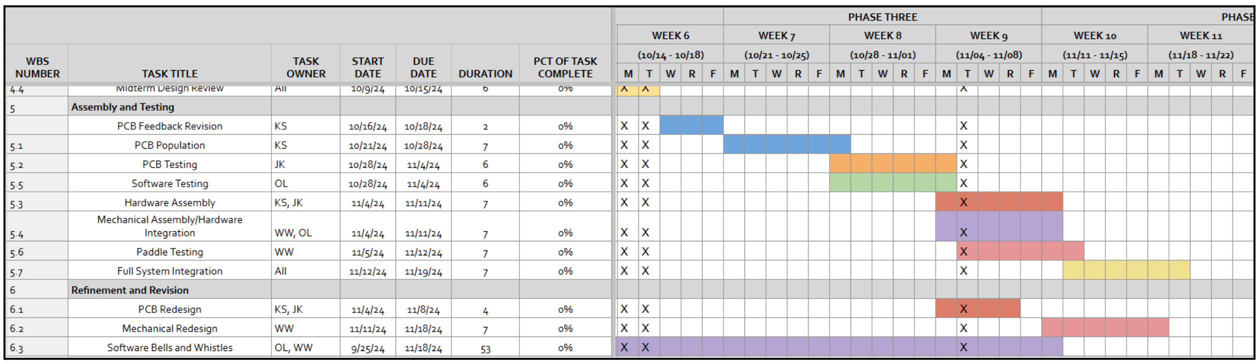

Fig. 36: Fourth Section of Original Gantt Chart

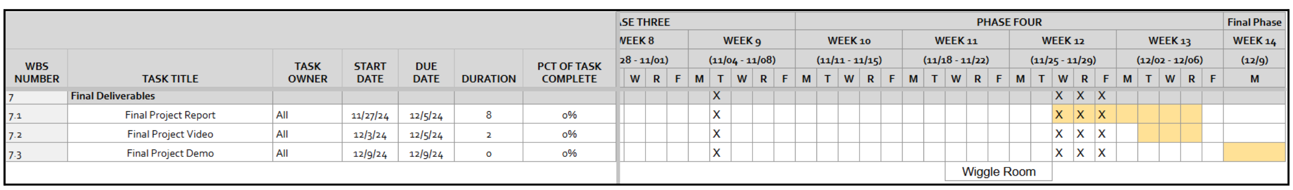

Fig. 37: Fifth Section of Original Gantt Chart

The final Gantt chart is much different than the original chart. This is primarily due to the sensor design change from FSR's to piezo's during week nine. Due to this change, the PCB design, revision, population, and testing were all extended later into the semester. Additionally, this delayed the ability to begin full system integration and testing. However, the team was able to handle the design change and the resulting delays it caused, and complete the project on time. The legend from the original Gantt chart is the same one used in the final Gantt chart. The final Gantt chart can be seen below in Figures 38-41.

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE
1	**Initial Concept and R&D**					
1.1	Feature Selection	All	9/6/24	9/13/24	7	100%
1.2	Define requirements and constraints	All	9/6/24	9/18/24	12	100%
1.3	Research Sensors	All	9/6/24	9/18/24	12	100%
1.4	System Block Diagram	KS	9/6/24	9/18/24	12	100%
1.5	IMU Algorithm Research	OL	9/6/24	9/18/24	12	100%
1.6	Paddle Design Research	WW	9/6/24	9/18/24	12	100%
1.7	BLE Central on PC Research	OL	9/17/24	9/24/24	7	100%
1.8	Develop Test Plan	All	9/6/24	9/18/24	12	100%
1.9	Project Proposal	All	9/16/24	9/20/24	4	100%

Fig. 38: First Section of Final Gantt Chart

The second section of our final gantt chart remained unchanged, as this section primarily focused on original part selection.

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE
3	**System Design**					
3.1	PCB Layout	KS, JK	9/18/24	10/11/24	23	100%
3.2	Learn ZephyrOS	WW	9/18/24	9/25/24	7	100%
3.3	Software Implementation	OL, WW	9/25/24	11/20/24	55	100%
3.4	Paddle Design	WW	9/23/24	10/4/24	11	100%
3.5	Handle Model	WW	10/2/24	10/23/24	21	100%
4	**Deliverables**					
4.1	Posters Due	All	9/23/24	9/29/24	6	100%
4.2	Poster Session	All	10/4/24	10/4/24	0	100%
4.3	Initial PCB Design Due	KS	10/11/24	10/14/24	3	100%
4.4	Midterm Design Review	All	10/9/24	10/15/24	6	100%

Fig. 39: Third Section of Final Gantt Chart

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE
5	**Assembly and Testing**					
5.1	FSR Testing	KS	10/9/24	10/15/24	6	100%
5.2	Handle Printing	WW	10/22/24	10/28/24	6	100%
5.3	2nd PCB Feedback Revision	KS	11/13/24	11/15/24	2	100%
5.4	2nd PCB Population	KS	11/14/24	11/18/24	4	100%
5.5	2nd PCB Testing	JK	11/15/24	11/19/24	4	100%
5.6	Piezo Testing	KS	11/8/24	11/13/24	5	100%
5.7	Software Testing	OL	10/28/24	11/28/24	30	100%
5.8	Hardware Assembly	KS, JK	11/20/24	11/26/24	6	100%
5.9	Mechanical Assembly/Hardware Integration	WW, OL	11/20/24	12/3/24	13	75%
5.10	Paddle Testing	WW	11/5/24	12/3/24	28	75%
5.11	Full System Integration	All	12/1/24	12/7/24	6	50%

Fig. 40: Fourth Section of Final Gantt Chart

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE
6	**Refinement and Revision**					
6.1	PCB Redesign	KS, JK	11/4/24	11/8/24	4	100%
6.2	Mechanical Redesign	WW	11/11/24	11/26/24	15	100%
6.3	Software Bells and Whistles	OL, WW	9/25/24	12/3/24	68	90%
7	**Final Deliverables**					
7.1	Final Project Report	All	11/27/24	12/5/24	8	70%
7.2	Final Project Video	All	12/3/24	12/9/24	6	0%
7.3	Final Project Demo	All	12/9/24	12/9/24	0	0%

Wiggle Room

Fig. 41: Fifth Section of Final Gantt Chart

Costs

For the project, the total cost came out to $499.37. The full expense breakdown is shown in Appendix A. The bulk of the cost is a result of the force sensors, IMU sensors, MCUs, PCBs, and paddles. These components alone contributed $290 to the expenses. Some mechanical materials like the printing filament and steel sheets also contribute significantly to the cost. The force sensors, included in the original design, were the most costly item, contributing $142 to the expenses. The final design includes piezo sensors, which accounted for $88. Through design changes and initial testing/research parts, the cost increased substantially. To consider the design changes and testing/research parts, a final expense breakdown was created to resemble the cost of just the final product. For example, this cost does not include the FSR sensors as they were not used in the final product. This expense breakdown is shown in Appendix B. To build the paddle with just the required parts, the cost came out to $237.32. This is substantially less than the actual cost, as the FSR sensors and other miscellaneous testing parts were not considered or used in the final product. Although the costs were high, the PIKL paddle remained under budget and could be less expensive if the paddle was to be built again.

The estimated cost before starting the project was $341. The breakdown of the estimated expenses is shown in Appendix C. This included many of the major components listed above as well as some miscellaneous costs. Comparing this to the actual expenses, PIKL was over budget. This is due to more money being spent on force sensors, paddles, and mechanical parts that were not originally considered. Comparing this to what the final product would cost if replicated, it came in under budget. This is due to the PCBs and force sensors costing less in the final design compared to the original design. Overall, the project's expenses could have been reduced by doing more research, testing, and designing of the components and parts that were ordered.

If PIKL were to be manufactured, it would cost approximately $240 per paddle, totaling 2.4 million dollars for 10,000 smart paddles. However, this cost can be reduced through bulk-order discounts from parts suppliers. For instance, sourcing components for 10,000 paddles from Digi-Key brought the total cost down to 1.8 million dollars, resulting in $600,000 in savings. A detailed breakdown of these savings is provided in Appendix D. When considering large-scale manufacturing of a product, this can be a substantial advantage in developing a successful and marketable product. Automation could help contribute to time savings, which could potentially contribute to cost savings. The PCB process, printing process, and assembly process could all be automated, albeit at a cost. Initially, there would be large upfront costs to make the development of PIKL automated, but in the long run, it could provide savings.

With automated equipment, more paddles can be made per day, which in turn allows more paddles to be sold. For PIKL, it would be ideal to have an automated process, as it would take a significant amount of time to make just one paddle and would increase product quality and consistency. Through an automated process, time can be saved, and in the future, costs can be saved.

Final Results

As discussed in the performance objective section and testing sections, the PIKL prototype provides the following pickleball metrics: swing speed calculation, impact force classification, impact location estimation, and stroke classification. The PIKL is housed in a custom handle that cleanly integrates into an existing pickleball paddle, maintaining an authentic playing experience. All metrics are visualized in real-time over BLE on the user's device in a GUI program. This interface also provides match history analysis to view progression over time. The final results for the PIKL can be derived from the results of the system integration testing. Overall, the PIKL accomplishes its goal of providing valuable and accurate pickleball metrics in a seamless playing experience. The accuracy results of each feature are summarized in Table 19 based on the results from the system integration section.

Feature	Result
Swing Speed	Accurate Swing Speed Differentiation
Impact Location	70% Accuracy
Stroke Classification	76.7% Accuracy
Impact Force	83.3% Accuracy

Table 19. Accuracy Results of Final PIKL Features

Shown below in Table 20 is the rubric we outlined for ourselves in the proposal.

Category	3 Points	2 Points	1 Point	0 Points
Impact Location	Consistently useful detection of impact location.	Generally useful impact detection, some inconsistencies.	Impact location occasionally useful, but often not detecting.	No useful impact detection.
Force Reading	Consistently useful force readings.	Generally useful force readings, some inconsistencies.	Force readings are rarely useful.	No useful force readings.
Swing Speed Reading	Consistently useful swing speed readings.	Generally useful swing speed readings, some inconsistencies.	Swing speed readings are rarely useful.	No useful swing speed readings.
Wireless Communication	Data is transmitted reliably with minor or no issues.	Communication works most of the time with occasional issues.	Communication is inconsistent, with delays or disconnection.	No wireless communication.
User Program	Program provides clear and useful feedback.	Program is useful but could use improvement.	Program is difficult to use or lacks information.	No user program.
Mechanical Build	Paddle is built well and is durable.	Paddle works with minor mechanical issues.	Paddle has major mechanical flaws that affect play.	Paddle is not functional or useful.

Table 20. Grading Rubric

Applying this rubric, we yield the following evaluation on the PIKL prototype:

+ 2 points: Three piezo sensors are able to locate where the impact was consistently, but some hits or locations are not the most accurate.

+ 3 points: The accelerometer is able to consistently determine impact force.

+ 3 points: MCU with external BLE antenna is able to reliably transmit data within a reasonable range.

+3 points: Seen in the GUI section, all our features are well-displayed, from live shot data to whole match data, providing the user good feedback.

+ 3 points: Custom handle was developed well with a focus on robustness and durability and enough space to contain all the components.

This sums to a total of 17 points which according Table 21 below, is an A. Overall, we sufficiently met all of our success criteria. The only minor issue with the final prototype is slightly lower accuracy of paddle impact location estimation, especially with shots that contact the top of the paddle face.

Letter Grade	Points
A+	18
A	16-17
B	13-15
C	9-12
D	1-8
F	0

Table 21. Grading Letter

Engineering Insights

This project presented many new learning opportunities, in regards to new technical skills. The hardware side used Multisim, LabVIEW, Waveforms, and KiCad as their primary tools. With previous experience in Multisim and Waveforms, these skills were enhanced and developed throughout the course of the project. LabView and KiCad were newer resources that needed to be learned. LabVIEW enabled the team to view similar data that Waveforms presented. KiCad enabled the team to layout and develop the PCB designs. Additionally, circuit design and sensor testing presented new opportunities as they were tailored specifically for this design and project. Soldering surface mounted components on the PCB was a skill that had to be developed for all the small resistors and capacitors we used.

From a software perspective, there were a lot of new skills to learn and develop. One super important skill is debugging. The group had prior course experience with debugging software, but this capstone definitely honed everyone's debugging skills as the software is completely custom and novel, with lots of potential issues that had to be discovered and addressed through debugging tools like VSCode's interactive debugging feature with hardware breakpoints.

The mechanical side used Fusion 360, Ultimaker Cura and Bambu Studio, and the Ultimaker S3 and Bambu P1S as their primary tools. There was very little prior experience besides some minor use of Inventor, so there was a big learning curve for all the tools. The most important tool, Fusion, has a learning depth far greater than the slicing tools (Cura/Studio) and the printers (S3/P1S), and it was not completely expected. In the future, with regard to new CAD software, the team will plan out a more appropriate timeline for learning the software. With that said, however, it is important to note that all these tools are learn-by-doing, one can only get so much from guides and videos. The best way to learn mechanical design software in-depth is by utilizing it.

Throughout this capstone project, the team learned a lot of valuable lessons from both a technical and time management perspective of the engineering process.. Effective time management was crucial, keeping the team on track and adapting when problems arose. For example, with five weeks left the group made a major design change regarding the sensors. This briefly provided a setback, as a new circuit had to be tested and designed, along with a new PCB design. Being able to accommodate and overcome these changes strengthened the team's time management skills.

One key insight was leveraging creative solutions by using software to address hardware issues, and vice versa. For instance, we encountered a hardware issue with reduced sensitivity in the piezo sensor at the top of the paddle. While this could have been resolved at the circuit level by adjusting resistor values, time constraints late in the semester made desoldering and testing impractical. Instead, we modified the software to scale the ADC value of this specific sensor, matching it with the others. Conversely, hardware optimizations also simplified and enhanced software performance. For example, the piezo sensors could reliably detect impacts, allowing the embedded software to bypass continuous signal processing on IMU data. Instead, the MCU collected and checked ADC data, triggering intensive IMU signal processing only when an impact was detected. This approach made the system more efficient while maintaining functionality.

Financial management taught the importance of budgeting and resource allocation, ensuring the project stayed within the set limits. Towards the end when the sensor type was switched, the budget had to be considered, as a large chunk was previously spent on the first sensor type. Additionally, teamwork and communication played pivotal roles throughout the project. This included communication amongst the team as well as communication with the advisor. Lastly, maintaining morale involved celebrating small accomplishments, providing constructive feedback, and fostering an open environment that kept the group motivated despite setbacks or disagreements. These lessons collectively contributed to a successful, enjoyable, and enriching capstone experience.

To future capstone students, the group's advice is to embrace the course from start to finish. With only one semester to complete the project, the time flies and is recommended to start early and stay on task. Time management is a crucial skill to have for the capstone project and in any future endeavors. Communication and teamwork is also a crucial skill for this course. Being able to communicate effectively and efficiently will make the project process smoother and build good team morale. This includes being open and respectful to others' views and opinions on the project decisions. Another tip is to have fun. The capstone will require many hours, but having a close-knit group you can joke and mess around with will make all the time spent on the project much more bearable. Additionally, do not be afraid of new or unknown tools, capstone is a great place to challenge yourself and learn a lot. Lastly, be prepared for setbacks and mistakes. There may be times when parts are missing or delayed, or times when a major design decision is made. Mistakes and setbacks present learning opportunities for the group to grow as engineers. Keep in mind, the project is not just about the final product, but rather the skills and lessons gained along the way.

Future Work

As a prototype, the PIKL is a success. However, there are still numerous areas for future improvement. For one, more reliable swing speed estimation using ground truth data. Currently, the PIKL estimates swing speed using a median filter on three different swing speed calculations. This method is effective in providing reliable and reasonable estimations of swing speed, but it is not reinforced with ground truth swing speed data. To improve the swing speed estimator, the actual speed of the swing could be measured using a radar gun and compared to the IMU measurements. This comparison can help to calibrate the IMU and the data processing code. Additionally, this ground truth data could be used to tune speed estimation and could even be incorporated into a supervised learning algorithm such as linear regression for more accurate measurements.

Secondly, the PIKL could be developed for more advanced stroke classification. Currently, the PIKL can accurately classify between forehands and backhands, but it is not perfect. Stroke classification could be improved to also classify overheads, volleys, serves, and dinks. Advanced stroke classification would require more complex signal processing and perhaps more advanced algorithms such as neural networks or Support Vector Machines (SVMs). One potential pitfall here is the amount of data needed to train such a model. Accurate classification models require lots of labeled data which this capstone didn't have the time to collect and label [45].

Further, another consideration is how much swing data from the gyroscope and accelerometer would be needed to input into a stroke classification model. If the amount of data is large (e.g 100s of samples from all three axes for both sensors), then this could bottleneck BLE transmission and cause increased power consumption and processing time. On the hardware side, the project could be expanded by implementing a recharging circuit for the 3.7 V lithium-ion battery. As of now, the battery is hidden inside the handle and inaccessible to the user. A recharging circuit that could be interfaced with a port on the handle would be a good expansion that would expand the practicality and long term use of the PIKL. One important consideration for adding a recharging circuit is the limited space inside the handle. The handle is already tight with a custom PCB, MCU, battery, and IMU sensor. This means any new recharging circuit would require a more compact PCB design and circuit layout that would still need to fit in the existing paddle handle.

A final hardware improvement would be better sensor selection and research. While the piezo sensors were the winners among the various sensors that were tested, they were not without drawbacks. The piezo sensors can be very delicate and once damaged (from tape or folding), they are practically useless. Furthermore, while paddle impact location is mostly successful on the PIKL, there is definitely room for improvement. The limiting factor in this feature was the reliability and consistency of the piezo sensors. In all, researching and selecting higher quality vibration or force sensors could improve the resilience and accuracy of impact location estimation. With additional time and resources, the PIKL could be improved with more reliable and accurate performance metrics, through those various design implementations and changes.

References

[1] Mackie, B. (2024, March 13). "Pickleball statistics - the numbers behind America's fastest growing sport in 2024," *Pickleheads*, Mar. 13, 2024. [Online]. Available: Pickleball Statistics - The Numbers Behind America's Fastest Growing Sport in 2024 | Pickleheads

[2] DeMelo, J. (2022, September 3). "Why is pickleball so popular?," *The New York Times*, Sep. 3, 2022. [Online]. Available: https://www.nytimes.com/2022/09/03/well/move/pickleball-popular-sport.html

[3] Frevel, N., Beiderbeck, D., & Schmidt, S. L. (2022). "The impact of technology on sports – A prospective study," *Technological Forecasting and Social Change*, vol. 182, p. 121838, 2022. [Online]. Available: https://doi.org/10.1016/j.techfore.2022.121838

[4] Marsh, J. T. (1993). "Golf swing measurement system," U.S. Patent 5,257,084 A, Nov. 2, 1993. [Online]. Available: https://patents.google.com/patent/US5257084A/en

[5] Cavallaro, R. H., Steinberg, E., Gueziec, A., & Mozes, A. (2008). "Virtual strike zone," U.S. Patent 7,341,530 B2, Mar. 11, 2008. [Online]. Available: https://patents.google.com/patent/US7341530B2/en

[6] VanOs, D. (2024, February 2). "What is the sweet spot on a pickleball paddle?," *Selkirk Sport*, Feb. 2, 2024. [Online]. Available: https://www.selkirk.com/blogs/paddles-and-product-education/what-is-the-sweet-spot-on-a-pickleball-paddle

[7] TE Connectivity, "FDT Series Elements with Lead Attachment," Data Sheet, Accessed: Dec. 3, 2024. [Online]. Available: https://www.te.com/commerce/DocumentDelivery/DDEController?Action=showdoc&DocId=Data%20Sheet%7FFDT_Series%7FA1%7Fpdf%7FEnglish%7FENG_DS_FDT_Series_A1.pdf%7F11031269-00

[8] S. T. Thompson, "Overview of Piezoelectric Sensors," Defense Technical Information Center, ADA429659, Accessed: Dec. 3, 2024. [Online]. Available: https://apps.dtic.mil/sti/pdfs/ADA429659.pdf

[9] Texas Instruments, "Application Report: Understanding Microcontroller Peripherals," Texas Instruments, Accessed: Dec. 2, 2024. [Online]. Available: https://www.ti.com/lit/an/sloa033a/sloa033a.pdf

[10] C. Castellanos, "Piezo Vibration Sensors," Physical Computing, Accessed: Dec. 3, 2024. [Online]. Available: https://physicalcomputing.ccastellanos.com/labs/sensors/piezo-vibration-sensors/

[11] Particle, "Argon Datasheet," Particle, Accessed: Dec. 3, 2024. [Online]. Available: https://docs.particle.io/reference/datasheets/wi-fi/argon-datasheet/

[12] Nordic Semiconductor. "nRF52840 Product Specification v1.11." Nordic Semiconductor, Accessed: Dec. 6, 2024. [Online]. Technical Manual PDF.

[13] SparkFun Electronics, "ISM330DHx Datasheet," SparkFun Electronics, Accessed: Dec. 3, 2024. [Online]. Available: https://cdn.sparkfun.com/assets/d/4/6/d/f/ism330dhcx_Datasheet.pdf

[14] SparkFun Electronics, "Product page for ISM330DHCX: MEMS IMU Breakout," SparkFun Electronics, Accessed: Dec. 3, 2024. [Online]. Available: https://www.sparkfun.com/products/19895

[15] H. Zhang, X. Wang, Z. Guo, Y. Qian, Y. Shang, and D. Cai, "On Impact Damage and Repair of Composite Honeycomb Sandwich Structures," *Materials*, vol. 16, no. 23, 2023, doi: 10.3390/ma16237374.

[16] Pickleball Science. "Thermoformed Paddle Delamination Revisited," Pickleball Science, Accessed: Dec. 6, 2024. [Online]. Available: https://pickleballscience.org/thermoformed-paddle-delamination-revisited/

[17] 3DMaker Engineering. "3D Printer Filament Guide," 3DMaker Engineering, Accessed: Dec. 6, 2024 [Online]. Available: https://www.3dmakerengineering.com/blogs/3d-printing/3d-printer-filament-guide

[18] SparkFun Electronics, "Board dimensions for ISM330DHCX: MEMS IMU Breakout," SparkFun Electronics, Accessed: Dec. 5, 2024. [Online]. Available: https://cdn.sparkfun.com/assets/4/a/f/8/9/19895_9DoF_BoardOutline.png

[19] SparkFun Electronics. "400mAh Lithium Ion Battery Datasheet." SparkFun Electronics, Accessed: Dec. 5, 2024. [Online]. Available: https://cdn.sparkfun.com/datasheets/Prototyping/spe-00-502535-400mah-en-1.0ver.pdf

[20] McMaster-Carr. "Passivated 18-8 Stainless Steel M2 Phillips Flat Head Screw." McMaster-Carr, Accessed: Dec. 6, 2024. [Online]. Available: https://www.mcmaster.com/92010a111/

[21] McMaster-Carr. "Steel M2 Pan Head Phillips Screw." McMaster-Carr, Accessed: Dec. 6, 2024. [Online]. Available: https://www.mcmaster.com/92005a016/

[22] McMaster-Carr. "Steel M3 Phillips Flat Head Screws." McMaster-Carr, Accessed: Dec. 6, 2024. [Online]. Available: https://www.mcmaster.com/91420a130/

[23] McMaster-Carr. “Aluminum M3 Heat-Set Inserts for Plastic." McMaster-Carr, Accessed: Dec. 6, 2024. [Online]. Available: https://www.mcmaster.com/94459a421/

[24] McMaster-Carr. "Aluminum M2 M3 Heat-Set Inserts for Plastic." McMaster-Carr, Accessed: Dec. 6, 2024. [Online]. Available: https://www.mcmaster.com/94459A418/

[25] Adam-Tech. "SW-R3-1A-A-XX-0 Data Sheet." Adam-Tech, Accessed: Dec. 5, 2024. [Online]. Available: https://app.adam-tech.com/products/download/data_sheet/203478/sw-r3-1A-a-XX-0-data-sheet.pdf

[26] Vishay Intertechnology. "TLHG440 LED Datasheet." Vishay Intertechnology, Accessed: Dec. 6, 2024. [Online]. Available: https://www.vishay.com/docs/83006/tlhg440.pdf

[27] Microchip Technology. "MCP6001/1R/1U/2/4 1 MHz Low-Power Op-Amp Datasheet." Microchip Technology, Accessed: Dec. 6, 2024. [Online]. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/MCP6001-1R-1U-2-4-1-MHz-Low-Power-Op-Amp-DS20001733L.pdf

[28] Nordic Semiconductor. "nRF52840 Product Specification: Programmable Peripheral Interconnect (PPI)." Nordic Semiconductor, Accessed: Dec. 6, 2024. [Online]. Available: https://docs.nordicsemi.com/bundle/ps_nrf52840/page/ppi.html

[29] H. Zhao, S. Wang, G. Zhou, and W. Jung, “TennisEye: tennis ball speed estimation using a racket-mounted motion sensor,” in *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*, Montreal Quebec Canada: ACM, Apr. 2019, pp. 241–252. doi: 10.1145/3302506.3310404.

[30] “A comparison of 802.11ah and 802.15.4 for IoT - ScienceDirect.” Accessed: Dec. 06, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405959516300650

[31] “bleak — bleak 0.22.3 documentation.” Accessed: Dec. 06, 2024. [Online]. Available: https://bleak.readthedocs.io/en/latest/

[32] Mouser Electronics. "FLX Series Flexible Cable Datasheet." Mouser Electronics, Accessed: Dec. 6, 2024. [Online]. Available: https://www.mouser.com/datasheet/2/1460/FLX_Datasheet_A301_RevI-3084724.pdf

[33] NextPCB, "IPC-2221: The Standard for Printed Circuit Board Design," NextPCB, Accessed: Nov. 30, 2024. [Online]. Available: https://www.nextpcb.com/blog/ipc-2221

[34] IPC, "IPC-2221: Generic Standard on Printed Board Design," Accessed: Nov. 30, 2024. [Online]. Available: https://www-eng.lbl.gov/~shuman/NEXT/CURRENT_DESIGN/TP/MATERIALS/IPC_2221.pdf

[35] USA Pickleball, "Equipment Standards Manual," USA Pickleball, Accessed: Nov. 30, 2024. [Online]. Available: https://usapickleball.org/docs/eec/Equipment-Standards-Manual.pdf

[36] Federal Communications Commission, "47 CFR § 15.247 - Operation within the bands 902-928 MHz, 2400-2483.5 MHz, and 5725-5850 MHz," Accessed: Nov. 30, 2024. [Online]. Available: https://www.ecfr.gov/current/title-47/chapter-I/subchapter-A/part-15/subpart-C/subject-group-ECFR2f2e5828339709e/section-15.247

[37] Bluetooth SIG, "Core Specification | Amended Version 5.4," Bluetooth SIG, Accessed: Nov. 30, 2024. [Online]. Available: https://www.bluetooth.com/specifications/specs/core-specification-amended-5-4/

[38] IEEE, "IEEE 802.15.4 Applications," IEEE, Accessed: Nov. 30, 2024. [Online]. Available: https://mentor.ieee.org/802.15/dcn/14/15-14-0226-00-0000-802-15-4-applications.pdf

[39] IEEE, "Bluetooth Low Energy Mesh Networks: A Standards Perspective," IEEE, Accessed: Nov. 30, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9071998

[40] Google Patents, "US Patent No. 20210252356A1," Accessed: Nov. 19, 2024. [Online]. Available: https://patents.google.com/patent/US20210252356A1/en

[41] Google Patents, "US Patent No. 7891231B2," Accessed: Nov. 19, 2024. [Online]. Available: https://patents.google.com/patent/US7891231B2/en

[42] Google Patents, "US Patent No. 11452919B2," Accessed: Nov. 19, 2024. [Online]. Available: https://patents.google.com/patent/US11452919B2/en

[43] Killshot Pro, "Killshot Pro Official Website," Killshot Pro, Accessed: Nov. 19, 2024. [Online]. Available: https://killshotpro.com

[44] Potenza Pickleball, "Smart pickleball paddle with performance analytics," Patent Pending. Accessed: Nov. 19, 2024. [Online]. Available: https://potenzapickleball.com/products/smartpaddle-by-potenza.

[45] M. Seyedan and F. Mafakheri, "Predictive big data analytics for supply chain demand forecasting: methods, applications, and research opportunities," Journal of Big Data, vol. 7, no. 1, p. 53, 2020, Accessed: Nov. 30, 2024. [Online]. Available: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-00327-4

Appendix

Appendix A: Full Expense Breakdown

Index	Item	Manufacturer #	Qty	Cost
1	Piezo Sensor	1-1002785-1	9	$88.05
2	FSR Sensor #1	FSR05BE	2	$5.58
3	FSR Sensor #2	A301-1	7	$95.34
4	IMU Sensor	SEN-19895	1	$39.95
5	IMU Jumper Connector	CAB-17261	2	$3.20
6	Microcontroller	ARGN-H	1	$27.52
7	AmazonPaddles	UP-01	2	$47.96
8	ASA Filiment	-	1	$20.99
9	Op-Amp Socket 8 Pin	A 08-LC-TT	5	$1.15
10	Op-Amp Socket 14 Pin	A 14-LC-TT	5	$1.65
11	Op-Amp 8 Pin	MCP6002-I/P	2	$0.88
12	Op-Amp 14 Pin	MCP6004-I/P	2	$1.18
13	Battery Connector Cable	4714	2	$1.90
14	Li-Ion Battery	PRT-13851	2	$9.90
15	Battery Connector	S2B-PH-K-S (LF)(SN)	11	$0.43
16	Large Switch	RA1113112R	1	$0.69
17	IMU QWICC Connector	PRT-14417	5	$2.85
18	IMU QWICC Cable	PRT-17260	2	$2.12
19	Mounted Buzzer	PT-1307	1	$0.88
20	Wired Buzzer	PT-1704	1	$0.83
21	MCU Socket 12 Pin	PPTC121LFBN-RC	6	$5.28
22	MCU Socket 16 Pin	PPTC161LFBN-RC	6	$7.14
23	Male Test Headers	61300211121	47	$4.84
24	Ceramic 47pF 603	CL10C470JB8NNNC	45	$0.71
25	Ceramic 0.01uF 402	CL05B103KB5NNNC	5	$0.50
26	Ceramic 1uF 805	CL21B105KBFNNNG	5	$0.55
27	Small Switch	SW-R3-1A-A-1-0	1	$0.27

Fig. 42: Expense Breakdown of PIKL Paddle 1-27

28	Head Screws #1	91420A130	1	$4.80
29	Inserts #1	94459A421	1	$4.73
30	Head Screws #2	92005A016	1	$5.00
31	Inserts #2	94459A418	1	$4.86
32	Head Screws #3	92010A111	1	$9.31
33	Ceramic 1uF 603	CL10B105KP8NNNC	6	$0.04
34	Ceramic 0.01uF 603	CL10B103KB8NNNC	16	$0.83
35	SM 100 ohm 603	ERJ-3EKF1000V	6	$0.59
36	SM 3.3k ohm 603	RC0603FR-073K3L	10	$0.12
37	SM 39k ohm 603	RC0603FR-0739KL	20	$0.91
38	1.2M ohm	CFR-25JB-52-1M2	25	$1.13
39	1.5M ohm	RNF14FTD1M50	25	$1.10
40	2M ohm	CF14JT2M00	25	$0.69
41	Silicone Sealant	76435T21	1	$5.38
42	Aluminum Sheet	2471T92	1	$7.91
43	Zener Diode 3.3V	BZB84-C3V3,215	5	$1.10
44	Steel Sheet	5303T121	1	$16.71
45	SM 39k ohm 805	RMCF0805FT39K0	12	$0.25
46	SM 41.2k ohm 1005	M55342K03B41E2RS2	6	$8.82
47	PCB V1	-	5	$10.59
48	PCB V2	-	5	$21.12
49	JTAG #1	-	1	$4.20
50	JTAG #2	-	1	$10.52
51	Dupont Cables	-	1	$6.31
Total Spent				**$499.37**
Total Left				**$0.63**

Fig. 43: Expense Breakdown of PIKL Paddle 28-51

Appendix B: Final Paddle Expense Breakdown

Final Parts Needed	Qty	Cost (Final)
Piezo Sensor	3	$20.00
IMU Sensor	1	$40.00
Microcontroller	1	$28.00
Paddles	1	$24.00
Op-Amp Socket 14 Pin	1	$0.35
Op-Amp 14 Pin	1	$0.60
Battery Connector Cable	1	$0.95
Li-Ion Battery	1	$5.00
Battery Connector	2	$0.45
IMU QWICC Connector	1	$0.60
IMU QWICC Cable	1	$1.10
MCU Socket 12 Pin	1	$0.90
MCU Socket 16 Pin	1	$1.20
Male Test Headers	7	$0.70
Small Switch	1	$0.30
Ceramic 1uF 603	1	$0.01
Ceramic 0.01uF 603	5	$0.25
Silicon Sealent	1	$5.40
Zener Diode 3.3V	2	$0.45
Steel Sheet	1	$16.70
SM 41.2k ohm 1005	4	$6.00
PCB V2	1	$21.00
JTAG #2	1	$10.50
ASA Filiment	1	$21.00
Insets #1	1	$4.70
Inserts #2	1	$4.90
Screws #2	1	$5.00
Screws #3	1	$9.30
Green LED	1	$0.16
Resistor 56k ohm	4	$0.40
Resistor 1.2k ohm	4	$0.40
Jumper cables	10	$2.00
Electrical Tape	1	$5.00
	Estimated Total	**$237.32**

Fig. 44: Final Expense Breakdown of PIKl Paddle

Appendix C: Estimated Expenses (before building the project)

Index	Item	Qty	Cost
1	Test Force Sensors	4	$18.00
2	Microcontroller	1	$40.00
3	IMU	1	$28.00
4	Force Sensors	6	$100.00
5	Op Amps	2	$2.00
6	Test Paddles	4	$24.00
7	PCB	2	$60.00
9	QWICC Cable	2	$4.00
10	Power/Battery	1	$20.00
11	Misc. Mechanical	1	$30.00
12	Misc. Electrical	1	$15.00
Total			**$341.00**

Fig. 45: Estimated Expenses

Appendix D: Large Quantity Expenses

Final Parts Needed	Qty large	Cost (Digikey)
Piezo Sensor	30000	$485,000.00
IMU Sensor	10000	$400,000.00
Microcontroller	10000	$25,200.00
Paddles	5000	$120,000.00
Op-Amp Socket 14 Pin	10000	$1,350.00
Op-Amp 14 Pin	10000	$4,400.00
Battery Connector Cable	10000	$9,500.00
Li-Ion Battery	10000	$55,000.00
Battery Connector	20000	$815.00
IMU QWICC Connector	10000	$5,600.00
IMU QWICC Cable	10000	$10,600.00
MCU Socket 12 Pin	10000	$3,900.00
MCU Socket 16 Pin	10000	$4,900.00
Male Test Headers	70000	$4,550.00
Small Switch	10000	$1,750.00
Ceramic 1uF 603	10000	$56.00
Ceramic 0.01uF 603	50000	$145.00
Silicon Sealent	5000	$27,000.00
Zener Diode 3.3V	20000	$600.00
Steel Sheet	7500	$125,250.00
SM 41.2k ohm 1005	40000	$27,600.00
PCB V2	10000	$210,000.00
JTAG #2	10	$105.00
ASA Filiment	2500	$52,500.00
Insets #1	5000	$23,500.00
Inserts #2	5000	$24,500.00
Screws #2	5000	$25,000.00
Screws #3	5000	$46,500.00
Green LED	10000	$1,150.00
Resistor 56k ohm	40000	$193.00
Resistor 1.2k ohm	40000	$267.00
Jumper cables	10000	$19,500.00
Electrical Tape	5000	$25,000.00
	Estimated Total	**$1,741,431.00**

Fig. 46: Expenses for 10,000 Paddles

Appendix E: PCB Schematic

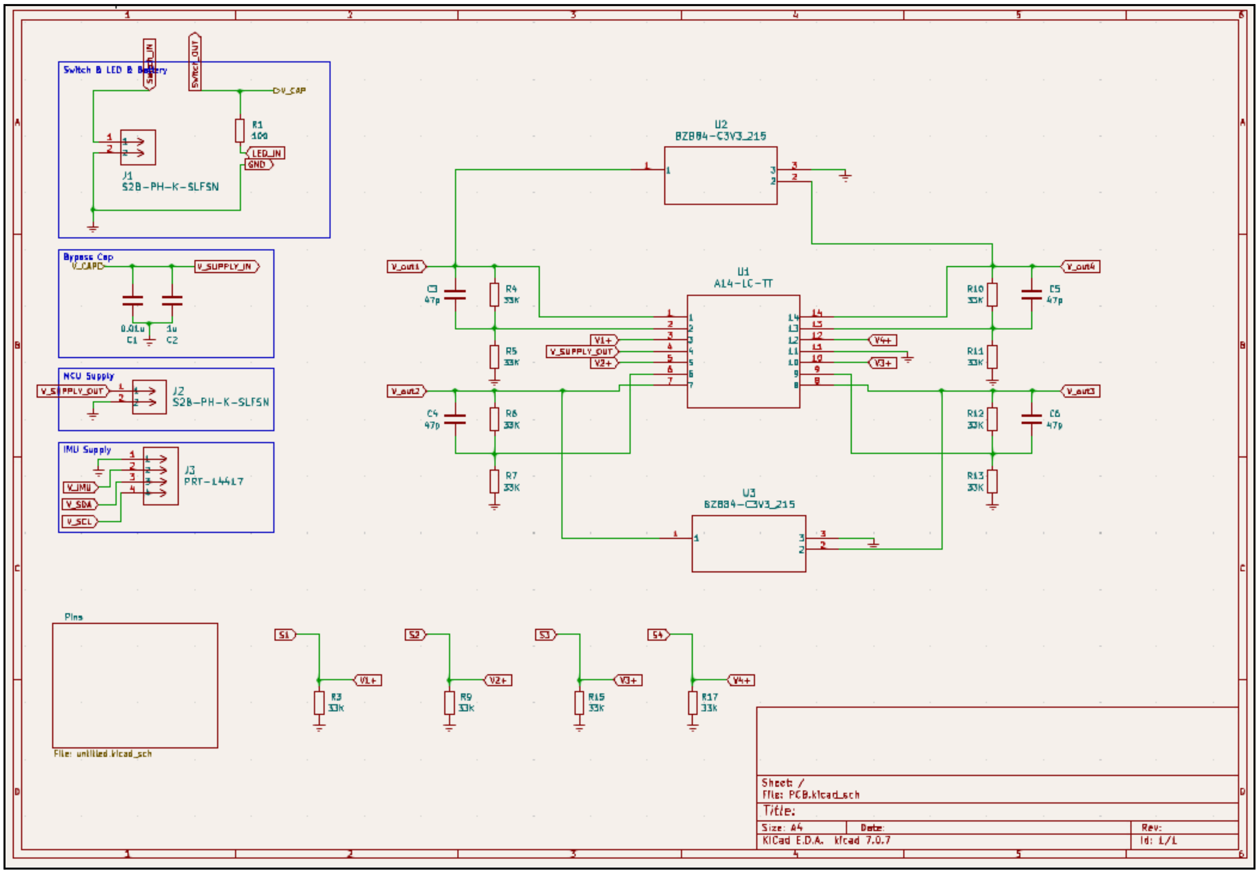

Fig. 47: PCB Circuit Diagram

Appendix F: PCB Schematic

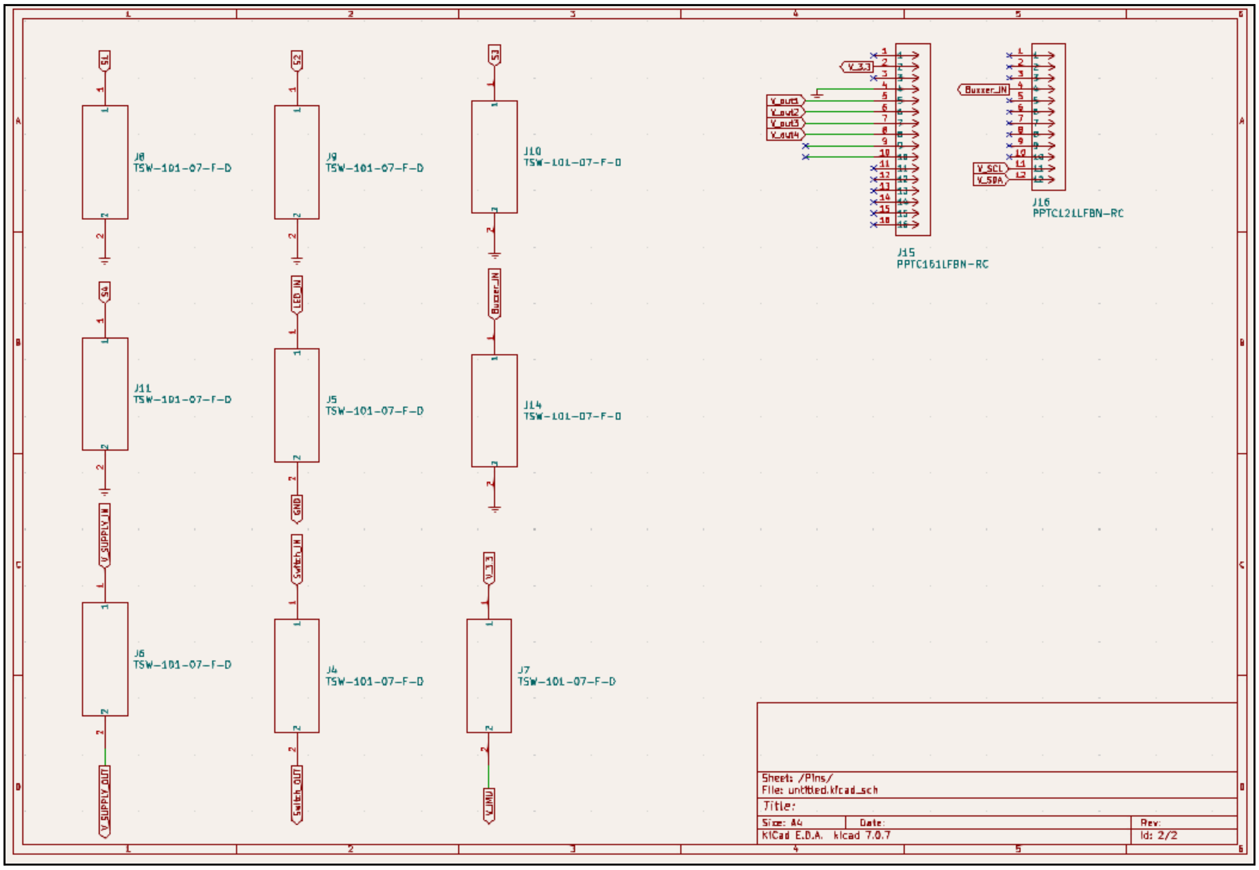

Fig. 48: PCB Circuit Diagram