SENSOR-INTEGRATED PUTTER

By

Kyle Smith

Mithesh Balle

Nathan Hwang

Final Report for ECE 445, Senior Design, Spring 2026

TA: Abdullah Alawad

6 May 2026

Project No. 74

Abstract

PutterIQ is a sensor-integrated putter system designed to measure putting stroke mechanics and transmit performance data to a mobile application in real time. The system combines custom embedded sensing hardware, including an IMU and piezo impact sensors, with wireless communication, frontend visualization, and camera-based ball analysis. The primary functions of the system are to determine stroke face angle, timing, and impact characteristics while preserving a natural putting stroke. Verification results showed the system achieved an average face-angle error of approximately ±0.54°, satisfying the ±0.6° face-angle accuracy target, and provided high-confidence impact detection and impact-location classification through fused IMU and piezo processing. The system also demonstrated reliable wireless stroke transmission to the mobile app and end-to-end real-time feedback during live practice testing. The final prototype successfully integrated sensing, embedded processing, and frontend analysis into a competition-ready smart putter platform. The project advanced to the final round of competition and placed 2nd out of 103 teams.

Contents

1. Introduction

1.1 Problem

Putting demands a high level of repeatability and consistency, where even small changes in club speed, face angle, and tempo can significantly affect the ball's direction and distance. For that reason, the average golfer tends to lose the most strokes during a round on the green. Hitting the ball far is not what separates strong golfers from the rest; consistently rolling the ball into the hole within par is much more difficult. During practice, it is often hard for golfers to notice small mechanical differences in their stroke, making it difficult to understand why one putt drops while another misses. Because putting relies on subtle and highly controlled motion, players frequently depend on subjective feel rather than measurable evidence, which leads to inconsistency and makes the root cause of errors difficult to identify. This extends beyond performance as well. Golf is played by millions worldwide and supports a multibillion-dollar industry, yet every little bit of the sport often comes at a high price. Poor putting consistency drives demand for private instruction, premium training aids, and expensive equipment that may be inaccessible to many golfers. An affordable feedback system that provides objective performance data would help make putting analytics more accessible and reduce the barrier to entry and improvement.

1.2 Solution

We developed the PutterIQ to address this problem. The PutterIQ is a sensor-integrated putter system that captures and analyzes stroke mechanics in real time. The system is designed to provide golfers with objective feedback on the physical aspects of their putting stroke that are otherwise difficult to perceive during practice. Rather than relying only on feel, users can review measured performance data to better understand why putts succeed or fail. At a high level, the system combines embedded sensing hardware mounted on the putter, wireless communication, a mobile frontend for live feedback, and a camera-based ball analysis pipeline. The onboard sensing hardware records club motion and impact behavior during each stroke, while the software processes this data into meaningful metrics such as face angle, tempo, impact timing, stroke speed, and impact location. In parallel, the camera subsystem analyzes the resulting ball path to quantify rollout behavior. Together, these subsystems provide a complete picture of how the user moved, what the putter did, and how the ball responded. The final result is an integrated training tool intended to make putting feedback more measurable, accessible, and actionable for golfers during practice.

1.3 High Level Block Diagram

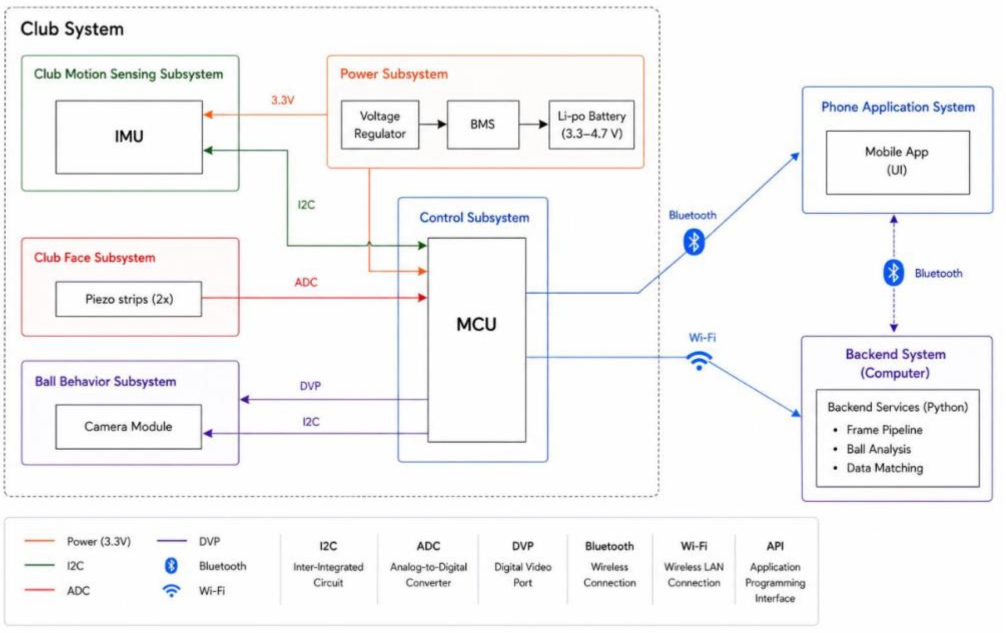


Figure 1. High-level block diagram for the sensor-integrated putter system, including the club sensing, control, power, mobile application, and backend analysis subsystems, along with the associated communication interfaces

2 Design

The final PutterIQ design evolved into a six subsystem architecture consisting of the impact and club-motion sensing subsystem, the MCU/Bluetooth/Wi-Fi communication subsystem, the data processing and mobile application subsystem, the camera subsystem, the power subsystem, and the club mounting and integration hardware. This differs from the original seven subsystems. Although the original concept treated club-motion sensing and impact sensing as more distinct functions, the final implementation merged them into a single sensing subsystem that fused IMU and piezo data. In this architecture, the IMU serves as the primary source for stroke segmentation and coarse impact timing, while the piezo sensors are used mainly for heel/toe/center impact classification and, when sufficiently excited, for refining impact timing. In addition, the system architecture was expanded from a single control PCB to a two-board design so that sensing and Bluetooth tasks could be separated from the expanded camera and Wi-Fi tasks, allowing the final system to better accommodate memory constraints and dual wireless communication requirements.

2.1 Overall System Architecture

Fig. 1 shows the high-level architecture of the final PutterIQ system. The system was organized so that the putter-mounted hardware captured raw stroke data, the embedded controller managed sensing and wireless communication, and the frontend displayed processed stroke and ball metrics to the user. On the putter side, the electronics acquired IMU and piezo data, applied stroke-state logic, and transmitted raw stroke packets to the mobile application over Bluetooth. In parallel, the camera subsystem captured burst images of the ball and transferred them over Wi-Fi to a local computer, where Python/OpenCV processing was performed. The processed ball-analysis results were then transmitted from the local computer back to the mobile frontend over Bluetooth for visualization and stroke association. This partitioned architecture allowed the system to combine low-latency stroke telemetry with higher-bandwidth image transfer and external vision processing while keeping the user-facing interface centralized in the mobile application.

2.2 Impact & Club Motion Sensing Subsystem

The impact and club motion sensing subsystem was designed to capture both the overall motion of the putter throughout the stroke and the response of the putter head at impact. In the final implementation, this subsystem combined a BNO085 inertial measurement unit (IMU) with two piezoelectric strip sensors mounted on the out**er edges** of the putter face. Although the original architecture treated club-motion sensing and impact sensing as separate functions, testing showed that the two sensing paths were most effective when fused into a single subsystem. The IMU provided stroke-motion data and a consistent timing anchor, while the piezo sensors provided localized impact information that was useful for heel/toe/center classification and, when sufficiently excited, refinement of impact timing.

2.2.1 IMU Sensing

The IMU served as the primary motion sensor for the system. The BNO085 integrates a three-axis accelerometer, three-axis gyroscope, three-axis magnetometer, and onboard sensor-fusion processing in a single package, making it suitable for motion-tracking applications [1]. In the PutterIQ system, the IMU was used to measure the linear and rotational motion of the putter during the setup, backstroke, forward stroke, impact, and follow through portions of each putt. Because the IMU continuously measures club motion throughout the entire stroke at 200 Hz, it became the most reliable source for stroke motion and impact timing. Unlike the piezo sensors, which only respond strongly when sufficient impact energy is transferred into the putter face, the IMU provides usable data for every putt regardless of impact location or strike quality. This made it suited for detecting the beginning of backstroke, the transition into forward stroke, and the timing structure of the stroke. The raw IMU data is transmitted to the frontend, where it was processed into user-facing stroke metrics.

2.2.2 Piezo Impact Sensing

Two piezoelectric strip sensors were used to measure localized mechanical excitation in the putter head at impact. Piezoelectric film sensors generate an electrical signal when mechanically strained or deflected, making them well-suited for vibration and impact detection [2]. In this project, the two piezo strips were positioned so that their relative response could be used to determine whether the ball contacted closer to the heel, center, or toe of the putter face. In practice, the piezo sensors were most useful for impact-location classification and for fine timing refinement when a strong impact pulse was present, particularly because they were sampled at a much higher rate of 2 kHz. However, center strikes did not always excite the piezo sensors enough to serve as a reliable single timing source. This behavior was one of the main reasons the sensing architecture evolved toward IMU and piezo fusion rather than treating the piezo sensors as the primary impact detector.

2.2.3 Sensor Fusion Strategy

The final sensing strategy assigned different roles to the IMU and piezo sensors based on their respective strengths. The IMU was used as the primary source for continuous stroke motion and impact timing, while the piezo sensors were used mainly for heel/toe/center classification and, when a sufficiently strong response was present, refinement of the impact timestamp. In testing, center-contact putts often did not excite the piezoelectric strips enough to produce a reliable pulse, even though many of the intended stroke metrics depended on accurate impact location. However, when accelerometer data was plotted over time, a distinct impact spike was consistently observed when the club made contact with the ball during a stroke. This made the IMU a more reliable base timing anchor for the system, particularly in cases where the piezo response was weak. This design choice improved robustness by preventing center strikes or low piezo excitation from causing misalignment across the computed stroke metrics. When a sufficiently strong piezo impact pulse was present though, the piezo-derived impact time was used to refine and, when appropriate, override the IMU-based impact estimate, allowing the final impact timestamp to align more closely with the higher-rate contact signal. As a result, the final subsystem operated not as a set of independent sensors, but as a fused sensing architecture in which motion data and impact data complemented one another. This approach enabled the system to capture both the overall mechanics of the putting stroke and the localized contact characteristics of the putter head while maintaining reliable stroke detection across a wide range of putt conditions.

2.3 MCU, Bluetooth, and Wi-Fi Subsystem.

The MCU, Bluetooth, and Wi-Fi subsystem is responsible for coordinating stroke capture, the embedded control flow, and transmitting data to the appropriate downstream systems. In the final architecture, the putter-mounted MCU acquired raw IMU and piezo data, managed the putt-state logic, and transmitted stroke packets to the mobile application over Bluetooth. In parallel, the overall communication architecture also supported a separate camera-analysis path, in which a forward-stroke trigger was sent from the putter system to a local computer and the processed ball-analysis results were returned to the frontend. This subsystem therefore served as the bridge between low level sensing hardware and the higher level software analysis dataflow.

2.3.1 Embedded Control Architecture

The first putter-mounted MCU served as the control unit for the embedded sensing hardware. Its responsibilities included sampling the IMU and piezo sensors, maintaining the internal stroke-state logic, packaging raw stroke data into a structured packet format, and managing Bluetooth communication with the mobile frontend. Rather than computing the final user-facing performance metrics directly on the MCU, the embedded firmware captures and transmits raw stroke data. This reduced the computational burden on the embedded hardware and allowed the majority of the signal processing to be performed later in software, where algorithms could be refined more easily during development. The embedded firmware continuously monitored the IMU and piezo inputs but only recorded stroke data during a predefined putt window. This prevented continuous data capture and ensured that only the relevant stroke interval was packetized and transmitted. The final raw packet structure contained the necessary IMU samples, piezo samples, timing metadata, and session identifiers needed by the frontend to process the stroke to a human readable format. In this way, the MCU functioned primarily as the acquisition and transport layer between the club face mounted sensors and the data analysis step.

In the final system architecture, the MCU responsibilities were also separated across two hardware paths to not overexert the onboard PSRAM. In testing, Bluetooth and Wi-Fi transmission could not coexist on the same ESP32-S3 as they each used up too much memory to support the other. This resulted in the separation of the two. The first putter-mounted board handled motion sensing and Bluetooth stroke telemetry, while the second camera-side system handled burst image capture and Wi-Fi transfer. This design decision allowed the project to support both real-time stroke telemetry and higher-bandwidth burst image processing without exceeding the memory and communication constraints of a single-board implementation.

2.3.2 Putt Window and Finite State Machine

To make stroke capture reliable and manageable, the firmware a sensed putting stroke was detected around a dedicated finite state machine and an armed putt window. A physical button connected to GPIO 44 was used to arm the system before a putt. Once pressed, the MCU entered a ready state and began waiting for natural stroke motion that satisfies the state transition conditions. This explicit arming step simplified the sensing step by restricting actual putt data capture to the intended interval, rather than attempting to interpret arbitrary club motion continuously. The finite state machine defined the major phases of a putt as a sequence of states, including idle, ready, backstroke, forward stroke, impact, follow-through, and stroke complete, as can be seen in Figure 2. Transitions between these states were determined primarily from threshold-based analysis of the IMU signal (Table 1). This state-based structure made it easier to identify the beginning of backstroke, the transition into forward stroke, and the impact time region, while also providing a clear skeleton for deciding when to begin and end stroke capture. If the expected stroke progression did not complete properly, the system terminates the current window and classify the event as aborted rather than storing invalid or ambiguous data.

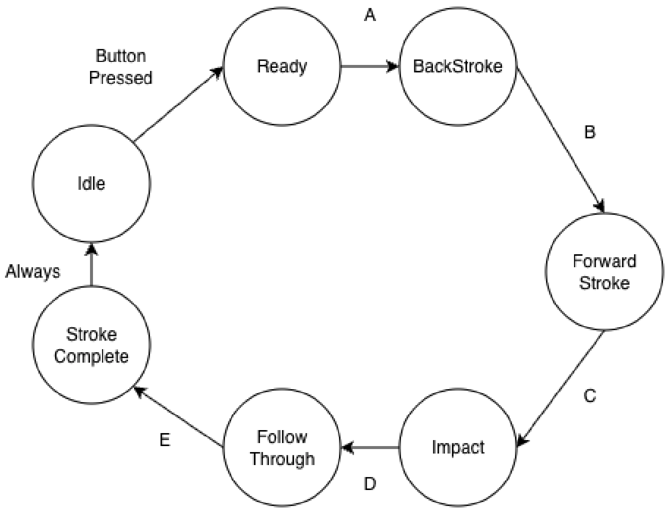


Figure 2: Firmware finite state machine used to manage the armed putt window, stroke-state transitions, and trigger generation for stroke capture and camera synchronization.

Transition	Requirements	What Happens?
Button Pressed	GPIO 44 Button Pressed	Putt Window Armed
A	Gyroscope Y axis > 12 °/sec for ≥3 IMU samples	Backstroke Timing Begins Calibration IMU Data Captured
B	Gyroscope Y axis transitions from > 0 °/sec to < 0 °/sec for ≥3 IMU samples	Backstroke Timing Ends Forward Stroke Timing Begins Camera Begins Burst Capture
C	Accelerometer Axis Delta ≥ 0.35 g	Impact Timestamped
D	Gyroscope remains < 0 deg/sec	Follow Through Timestamped

E	Gyroscope Y axis transition from < 0 °/sec to > 0 °/sec for ≥2 IMU samples	Putt Window Finished Bluetooth Data Sent to Frontend for Processing Wi-Fi Data Sent to OpenCV for Processing

Table 1. FSM Transition Requirements & Implications

The putt button window and finite state machine were especially important because they simplified several data processing tasks. First, they constrained the amount of raw data that needed to be stored and transmitted. Second, they provided the timing structure used later for frontend metric calculation. Third, they created a clean trigger point for the camera frame capture. In particular, the transition into forward stroke became the event used to generate the early trigger that initiated burst image capture on the camera side.

2.3.3 Wireless Communication and Triggering

Bluetooth is the primary wireless link between the physical putter and the mobile frontend. After the MCU captures a complete stroke within the putt window, the raw stroke packet transmits over Bluetooth to the mobile app. This path was chosen because it provides a direct, low-latency connection between the putter and the user-facing interface, making it suitable for live strokes and session based feedback. The firmware also emitted a small event message when forward stroke begins to be used to coordinate the camera subsystem. Once the frontend receives the trigger over Bluetooth, it forwards the event to the local computer responsible for burst image capture and OpenCV ball analysis. In this way, Bluetooth is not only used for stroke telemetry but also as the control link that initiates the higher-bandwidth camera workflow.

Wi-Fi is required for the camera side of the system because the burst image frames are too large to transmit efficiently over the same Bluetooth path used for the putter data. Instead, the camera board exposes the captured burst frames over a Wi-Fi connection to a local computer, where Python processing is performed. After the local computer completes the vision analysis, the final ball-path metrics are transmitted back to the mobile frontend over Bluetooth by advertising as a separate BLE peripheral. This communication architecture allows the system to combine a lightweight putter telemetry channel with a higher bandwidth burst image path, while still presenting the final results to the user within a single converged frontend interface.

Overall, the MCU, Bluetooth, and Wi-Fi subsystem provides the control and communication foundation for the project. It manages the timing of stroke capture, defines the finite state behavior of the putt window, transmits raw sensor data to the frontend, and coordinates the trigger and return path required for the external camera data analysis.

2.4 Data Processing & Mobile Application Subsystem

The data processing and mobile application subsystem is responsible for transforming raw sensor and camera data into user readable performance metrics. Rather than computing the final stroke metrics directly on the putter-mounted MCU, the system was designed so that the embedded hardware captures and transmits raw stroke data, while the frontend and associated software pipeline perform the actual data processing. This architecture reduces the computational burden on the embedded system, simplifies firmware packet generation, and allows the metric calculations to be updated more easily during development without repeated firmware changes. In the final implementation, the mobile application served as the central user interface for live feedback, session review, and data fusion.

2.4.1 Stroke Data Pipeline

The stroke data pipeline begins with the embedded system capturing raw IMU and piezo samples during the armed putt window. These samples are packetized on the MCU and transmitted to the mobile app over Bluetooth. Once received by the frontend, the raw data is parsed and processed into structured stroke information. This includes backstroke, forward, impact, and timing timestamps, impact location classification, and computation of the final user-facing metrics. In parallel, the frontend also acts as the bridge between the putter and the camera analysis. When the MCU detects the transition into the forward stroke, it transmitted a trigger event to the mobile application. The frontend then forwards this trigger over Bluetooth to the local computer running the Python/OpenCV burst analysis. After the camera processing is completed, the resulting ball behavior is transmitted to the frontend and associated with the corresponding stroke. This makes the mobile application the central converging point for both putter sensor and camera data.

2.4.2 IMU and Piezo Metrics

The putter sensor metrics are computed from the fused IMU and piezo data after the raw stroke packet arrives at the frontend. The IMU serves as the primary source for stroke timing structure, and key metrics like face angle, and stroke speed, and tempo, while the piezo sensors are used for impact location and refinement of impact timing when a strong pulse is present.

The stroke is segmented into setup, backstroke, forward stroke, impact, and follow through using threshold-based analysis of the IMU data, as described in Table 1. Timestamps are calculated from the indices associated with these motion windows. The backstroke duration, forward stroke duration, follow through duration, and total stroke duration are all computed from the segmented stroke intervals. Tempo ratio was calculated as the ratio of backstroke duration to forward-stroke duration

$$Tempo\ Ratio = \frac{t_{backstroke}}{t_{forward}} \tag{1}$$

The face angle at impact metric is also derived from the IMU data as the difference in roll angle between setup and that impact timestamp calculated before (Table 1):

$$\theta_{face} = \theta_{impact} - \theta_{setup} \tag{2}$$

Stroke Speed is estimated by using the accelerometer x and y values, compensating for gravity. We project the horizontal acceleration onto the dominant putt axis and integrate it over time to find linear velocity. We then align it to the timestamp to get our stroke speed at impact:

$$v(t) = \int a_{proj}(t)\, dt \tag{3}$$

Finally, Impact location is calculated using the heel and toe located piezoelectric strips. If one were to receive a signal > 300 ADC counts, then it is automatically placed there. If they both were to have a strong signal, the greater of the two would take it. Otherwise, the average response over the perceived window would be taken by each, and then the relative difference would be found and if the difference were > 120, then the higher average would be classified. Piezo imbalance can be calculated as:

$$\rho = \frac{Toe\ Avg - Heel\ Avg}{Toe\ Avg + Heel\ Avg + 1} \tag{4}$$

2.4.3 Camera Metrics

In addition to the stroke metrics derived from the putter sensors, the system also computes ball path metrics from the camera. After the camera burst frames are transferred over Wi-Fi to the local computer, Python/OpenCV processing is used to detect the ball center across the captured frame sequence and find a best fit path to the observed motion. From this path, the system calculates metrics such as path drift, RMS lateral deviation, and direction wobble.

Path drift is defined as the angle of the best fit ball path relative to the intended vertical roll direction. If the direction vector of the fitted line is (v_x, v_y) the drift angle is computed as:

$$\theta_{drift} = tan^{-1}\left(\frac{v_x}{v_y}\right) \tag{5}$$

To quantify straightness, the system measures the perpendicular displacement of the ball center from the best fit path and calculates the RMS lateral deviation:

$$\theta_{wobble} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(\theta_i - \theta_{drift})^2} \tag{6}$$

The Processed camera metrics are not computer on the mobile app but rather a local computer performing the computer vision analysis that are then transmits the resulting metrics back to the frontend over Bluetooth.

2.4.4 Mobile Visualization and Session Mapping

The mobile application presents the processed metrics to the user in both live and historical formats. During a session, the frontend displayed the most recent stroke metrics and associate plots of the raw or derived sensor signals. It also displays the processed ball path metrics returned from the camera pipeline. This gives the user immediate visual feedback on both stroke execution and ball response. For longer term review, the frontend stores each stroke locally in a maintained session history. In the final design, ball analysis results are mapped one-to-one to individual strokes so that a given stroke could be reviewed together with its corresponding ball behavior. This allows the session page to function not only as a list of captured putts, but also as a detailed record of stroke mechanics, impact characteristics, and resulting ball-path behavior.

2.5 Camera Subsystem

The camera subsystem captures burst images of the golf ball immediately after impact and transfers them to a local computer for vision analysis. In the final implementation, the subsystem uses an OV5640 camera module connected to a dedicated ESP32-S3 CB, which operates independently from the putter mounted sensing MCU. The camera board runs its own Wi-Fi access point named "OV5640-Burst" and exposes an HTTP based burst capture interface that the backend computer connects to directly.

The camera firmware maintains a short rolling buffer of approximately 350 ms in PSRAM. When the forward-stroke trigger is received, the firmware freezes the pre-trigger frames and continues capturing post-trigger frames for approximately 1400 ms at VGA resolution (640×480) with JPEG compression. The resulting burst of 40 frames is stored in PSRAM and made available for download through an HTTP manifest. This approach avoids continuous high bandwidth streaming and instead delivers a focused set of frames covering the contact and initial rollout window.

On the backend, two Python/OpenCV analysis scripts process the downloaded burst frames. The primary analyzer (`ball_analyzer.py`) implements a multi-stage detection pipeline. First, each frame is preprocessed with CLAHE histogram equalization and Gaussian blurring. Ball detection combines two sources, first OpenCV's HoughCircles transform operating on the equalized grayscale image, and white region contour analysis using an HSV saturation mask and brightness threshold. Each candidate is scored based on whiteness ratio, edge density in the perimeter ring, brightness contrast against the surrounding background, and positional continuity with prior frames. The highest scoring candidate is selected as the ball location. Once the ball is located, the Lucas-Kanade optical flow tracking is used to propagate both the ball position with the directional line orientation, reducing re detection cost and improving temporal consistency. The system also includes a low light adaptation mode that adjusts gamma correction, CLAHE clip limits, Canny thresholds, and scoring weights when the mean frame brightness falls below a configurable threshold.

From this orientation line, we are able to adjust and calculate orientation change between frames, which allows us to calculate our ball path metrics. Drift is calculated as the normalized arctan of our overall angular change off of the ideal roll.

2.6 Power Subsystem

The power subsystem supplies portable power to the embedded electronics while remaining compact enough to fit within the club-mounted enclosure. In this project, the primary design constraints for the power subsystem were battery size, weight, and compatibility with the operating voltage of the remaining electronics. Since the sensing and control electronics operate from a regulated 3.3 V rail, the system uses a 3.7 V Li-ion battery together with a voltage regulator to provide the required supply voltage. Battery selection was based primarily on practical integration requirements rather than aggressive runtime optimization. The battery must be large enough to power the system for extended testing and demonstration, while still fitting comfortably inside the enclosure and avoiding excessive added weight on the putter. As a result, the final battery choice balances usable capacity with compact form factor and acceptable mass. This simple power architecture is sufficient for the final prototype because it provides stable regulated power while meeting the packaging and usability constraints of the system.

2.7 Club Mounting & Integration

The club mounting and integration subsystem secures the physical components to the putter while preserving safe wiring, weight distribution, and a natural club feel. Since the final prototype includes multiple sensors, two MCU boards, wiring paths, a battery, and a mounted camera, physical integration plays a major role in determining whether the system remains usable as a putter rather than only as a breadboard prototype. A key consideration in the wiring layout is the separation of the piezo and I^2C signal paths. Since the piezo sensors produce sensitive analog signals while the IMU communicates over I^2C, the wiring is routed to reduce interference between these paths and to maintain signals. Routing considerations include feeding wires safely through the enclosure, providing access for USB debugging during development, and making sure that repeated assembly, disassembly, and testing do not place excessive strain on the electrical connections.

Mechanical modifications to the enclosure are also required to support the system layout. Openings are incorporated to provide a stable camera mounting position, wiring locations, and access points for debugging connections. The ready button is integrated near the handle so that the putt window can be armed conveniently without interfering with the normal grip. Throughout the integration process, battery placement, board placement, and wiring paths are chosen to keep the added mass low and to minimize disruption to the putter's balance and feel during use.

2.8 PCB Design

The final prototype uses two PCB designs distributed across the sensing and communication portions of the system. One PCB is mounted near the club head and supports the IMU, and piezoelectric sensing hardware used for club motion and impact detection. The second PCB design supports the MCU communication and control hardware. In the final prototype, two copies of the MCU PCB are used to support the split Wi-Fi and Bluetooth architecture of the system. This split approach allows sensing hardware to remain close to the putter head while separating the communication workload needed for Bluetooth and the Wi-Fi-based camera. The PCB layouts are also designed to support practical enclosure integration, wire routing, debugging access, and overall packaging constraints. Figures 3 and 4 show the final PCB layouts used in the prototype.

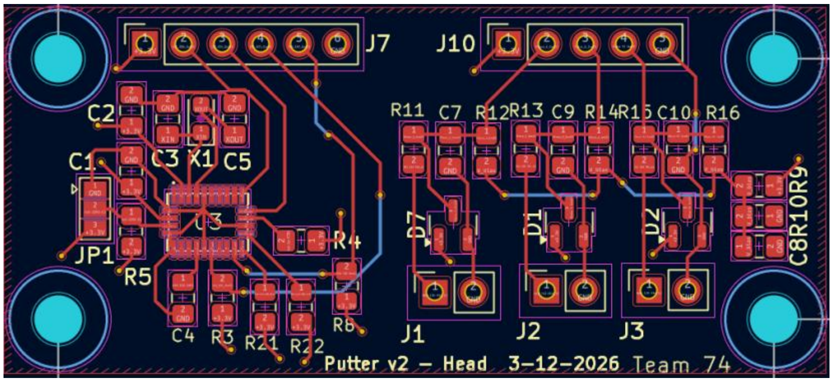


Figure 3. Club Head PCB Design

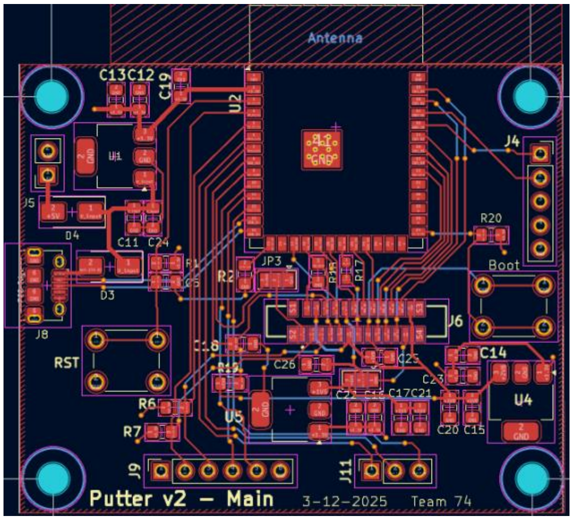


Figure 4. Club Shaft PCB Design (MCU PCB)

3. Design Verifications

This section evaluates the PutterIQ system against the project's high level performance requirements. The requirements we made for ourselves say that the putter must measure putting stroke face angle and timing with no more than ±0.6° face angle error and no more than 5 ms impact timing resolution when compared against high frame rate video. It must also transmit stroke data to the mobile application with 99% successful packet delivery, no repeated strokes, and no more than 500 ms end-to-end latency from impact to app display. Finally, the new components must add no more than 400 g to a standard mallet style putter without measurably interfering with a natural putting stroke feel. Verification of the face angle, timing, and impact location performance is presented in the impact and club motion sensing subsection, communication reliability and latency are evaluated in the MCU, Bluetooth, and Wi-Fi subsection, and physical integration such as weight and usability are addressed in club mounting subsection.

3.1 Impact and Club Motion Sensing Verification

Requirements:

- I2C communication between IMU and MCU with communication error < 2% per session.
- IMU measures linear acceleration along three axes with range $\geq \pm 8$ g and resolution ≥ 0.01 g.
- IMU measures angular velocity along three axes with range ≥ 500 °/s and resolution 0.1 °/sec.
- IMU measures putter head velocity with accuracy within ±5% relative to reference measurements.
- IMU measures face rotation with error, $\leq 0.6°$ for putts up to 10 feet.
- Motion tracking operates through the duration of the stroke without interruption or data loss $\geq 95\%$ of the time.
- Ball impact is detected ≥ 98% of the time during normal putting strokes.
- Ball impact is correctly classified (heel, center, or toe) with ≥ 90% accuracy.
- Impact timestamp captured within ~5 ms relative to true contact moment.
- Abnormal impact signatures (double hits, ground contact) are detected with ≥ 95% accuracy.
- Impacts are sampled at a minimum frequency of 2 kHz during the impact window.

3.1.1 Nine Axis IMU Communication and Motion Tracking Verification

To verify IMU communication and axis correctness, we performed repeated controlled stroke and orientation tests while monitoring the accelerometer, gyroscope, and magnetometer data received by the MCU and forwarded to the frontend. Each axis was checked by applying motion or rotation for which the expected dominant sensor response is known beforehand. Controlled translational motion and tilt were used to verify the accelerometer axes, while controlled club rotation across different directions was used to verify the gyroscope axes. The measured axis polarity, magnitude, and timing were then compared against slow motion video of the same motion to confirm that the correct axis responds with the correct sign and at the correct point in the stroke. Communication reliability was verified by confirming that the IMU data is read continuously throughout each stroke without missing samples, interrupted motion segments, or failed sensor reads. Across testing, no communication errors or data loss were observed, and the sensing pipeline continuously captures stroke data from setup to follow through. This satisfies the requirement that I2C communication remains reliable during a session and that motion tracking operate without interruption or data loss at least 95% of the time. In addition, the BNO085 configuration used in the system satisfies the required acceleration range of at least ±8 g and angular-rate range of at least ±500°/s [1].

3.1.2 Putter Head Velocity Verification

Putter head velocity is verified by comparing the IMU-derived stroke speed against a video reference measurement. The calculation is simply distance/time within the impact window. The measured velocity error remained within the required $\pm 5\%$ ramge The percent error is computed as:

$$\%\ error = \frac{|v_{measured} - v_{reference}|}{v_{reference}} * 100 \tag{7}$$

3.1.3 Face Rotation and Impact Timing Verification

Face Angle was verified using a repeated 10 foot putt test and comparing the system reported face angle against actual putt outcome and reference measurements. For each trial, the reported face angle at impact is checked against the expected launch direction window required for the ball to remain on a makeable line into the hole. Because the hole diameter is 4.25 in and the ball diameter is 1.68 in, the allowable center-offset window at the hole is limited, so small face-angle errors become meaningful over a 10 ft putt. Using this test setup, the system's reported face angle remains within an average error of ±0.54°, satisfying the required ±0.6° accuracy. Impact timing is verified against high-frame-rate video by comparing the detected impact instant to the reference impact frame. Using the IMU as the primary timing anchor and the piezo response for refinement when a strong local pulse is present, the final timing estimate remains within the required 5 ms resolution. The equation used to measure the offset distance is:

$$x = D * \tan(\theta) \tag{8}$$

Where x is the lateral offset at the hole, D is the putt distance and θ is the face angle error. This relation is used during the 10 foot putt test to interpret whether the reported angle stays within the acceptable make window. Against the videoed measurements, the findings can be seen in Figure 3.

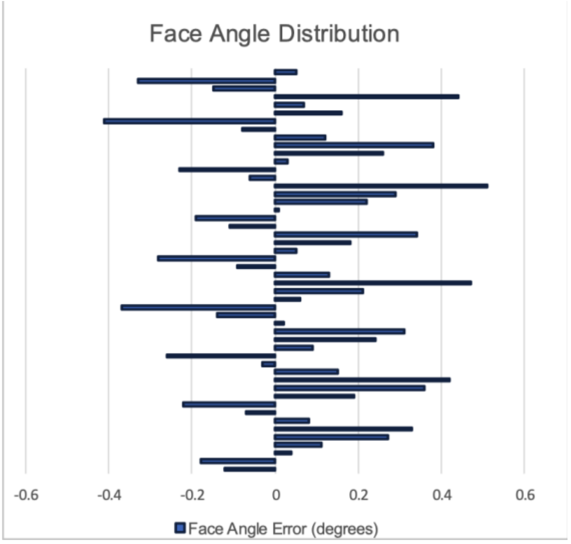
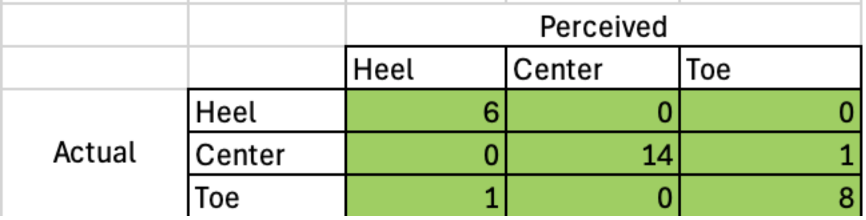


Figure 5. Face Angle error distribution

3.1.4 Impact Detection Verification

Impact detection was verified by comparing the sensed impact event against controlled putting trials and reference video. This verification evaluates impact detection rate, heel/center/toe classification accuracy, timestamp accuracy, abnormal-impact detection, and impact window sampling behavior. During testing, ball impact was detected in 100% of trials where a ball was present, satisfying the requirement of at least 98% detection during normal putting strokes. Impact location is classified as heel, center, or toe by comparing the relative piezo responses near the detected contact event, and the measured classification accuracy is 93% as shown in Figure 4, satisfying the required 90% minimum. Impact timing was verified against video by comparing the detected timestamp to the true contact moment, and the final impact timestamp remains within the required ~5 ms resolution. During the impact window, the piezo signals are sampled at 2 kHz, satisfying the required minimum sampling frequency. Abnormal impact signatures are also evaluated by examining cases such as double hits and ground contact. These events are identified by their distinct signal shape and timing behavior relative to a normal single-ball impact. The measured abnormal-impact detection accuracy satisfies the required 95% threshold.



		Perceived		
		Heel	Center	Toe
Actual	Heel	6	0	0
	Center	0	14	1
	Toe	1	0	8

Figure 6. Impact Location Confusion Matrix

3.2 MCU, Bluetooth, and Wi-Fi Verification

Requirements:

- Stroke metrics are computed within 500 ms of impact detection.
- Stroke data is transmitted for 100% of strokes.
- All local data and transmitted data are aligned 1-to-1
- Bluetooth connection stability maintained for at least 240 minutes.
- IMU and impact sensor data aligned within ±5 ms.

To verify the MCU and Bluetooth, we made sure that stroke metrics are computed and displayed within the required 500 ms window after impact detection (Figure 5), and stroke data is successfully transmitted for 100% of recorded strokes. In addition, all locally captured data and transmitted data remain aligned on a one-to-one basis, with no dropped, duplicated, or mismatched packets observed during testing. Bluetooth connection stability is maintained throughout extended operation exceeding the required 240 min duration, and the IMU and impact sensor data remain aligned within the required ±5 ms timing window. Together, these results verify that the embedded control and communication subsystem meets the required timing, transmission, and synchronization performance targets.

To verify our Wi-Fi subsystem, we ran our OV5460 Wi-Fi stream 3 separate instances for 4 hours continuously (same time as our battery subsystem testing). We verified no dropouts and consistent connectivity– that is we could connect with any device and follow the https link to correctly access the camera burst stream.

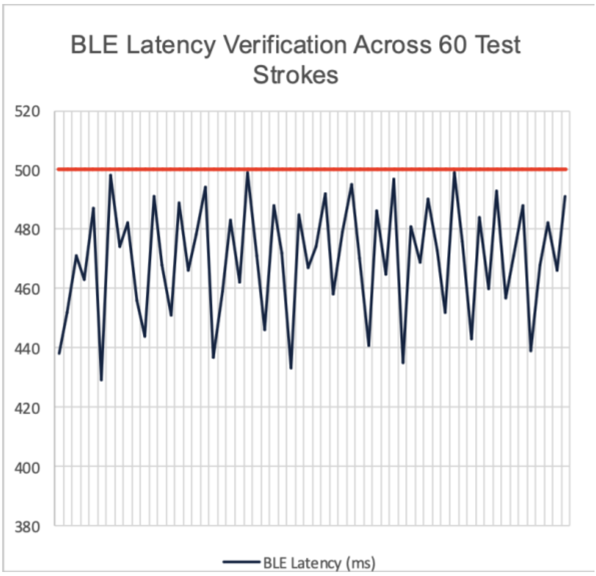


Figure 7. BLE Latency Test Results

3.3 Data Processing and Mobile Application Verification

Requirements:

- The app must display stroke metrics (speed, face angle, impact location, ball roll) within 2 seconds of putt.
- Stroke data must be stored in the database with 100% reliability and be retrievable for trend visualization.

This subsection verifies the responsiveness and storage reliability of the mobile application. During testing, the application displays stroke metrics, including stroke speed, face angle, impact location, and ball roll data, within 2 seconds of each putt. In addition, stroke data up to one hundred strokes is stored in the session database with 100% reliability and remains retrievable for later review until the next session. These results verify that the frontend meets the required live feedback and session-based data retention targets.

3.4 Camera Verification

Requirements:

- The camera captures video at a minimum rate of 20 frames per second.
- Subsystem detects the orientation line of the ball with ≥ 90% detection accuracy.
- Ball rotation error rate < 10% relative to reference video analysis.
- Wobble detected accurately when > 5° deviation from ideal roll.

To verify our camera subsystem, we created a timestamp-based frame tracker, and through 100 tests verified an average of 22.1 fps, which is greater than our minimum of 20 fps. Next, to test our orientation line, we tracked 30 consecutive putts and verified that 29.1/30 frames that contained a golf ball correctly identified the orientation line. We further used a protractor to verify that the orientation line of roll between frames was correct. To verify our rotation error rate, we used a protractor and verified our drift angles matched the actual drift of our ball path. Our average error rate was 3.2% off the actual drift, which satisfied our 5% marker. Lastly, we in 300 total frames of ball detection, we detected the ball in 299 frames, verifying our 98% metric.

3.5 Power and Integration Verification

To verify the power subsystem, we made sure to regulate the voltage stability and runtime required for normal system operation. During testing, the battery and regulation stage maintain a stable 3.3 V supply to all active components, with no observed power interruption during sensing, communication, or camera use. In addition, the selected battery powers the system for at least 4 h of continuous operation, satisfying the runtime target of approximately two rounds of use per charge.

3.6 Club Mounting & Integration Verification

For our mounting subsystem, it was important to keep the weight close to that of an actual putter. To do this, I selected a 1.3lb putter, and once gear was mounted, it weighed 1.95 lbs., adding 299 grams, satisfying our 400 gram requirement. We also mounted our electronics box high enough on the shaft that it centralized balance and ensured that the black box was not a hinderance in the putting view of the club head. Our second verification was that the “feel” of the club was correct. To do this, we surveyed 20 golfers, and received three 5’s, seven 4’s, eight 3’s, one 2, and one 1, satisfying our requirement of 18/20 >= 3.

4. Costs

4.1 Parts

Description	Manufacturer	Quantity	Extended Price	Link
IMU ACCEL/GYRO/MAG I2C 32BIT	Adafruit	1	$24.95	Link
SENSOR PIEZO FILM VIBRA TABS	TE Connectivity Measurement Specialties	3	$15.03	Link
ADAFRUIT OV5640 CAMERA BREAKOUT	Adafruit	1	$19.95	Link
ESP32-S3-WROOM-1-N8R8 DEV BRD	Espressif Systems	1	$15.00	Link
EMITTER IR 940NM 100MA RADIAL	Vishay Semiconductor Opto Division	2	$0.96	Link
IMU ACCEL/GYRO/MAG I2C 32BIT	CEVA Technologies, Inc.	1	13.05	Link
OV5640 CAMERA XIAO ESP32S3 SENSE	Seeed Technology Co., Ltd	1	$11.99	Link
RF TXRX MOD BT WIFI PCB TH SMD	Espressif Systems	1	$6.56	Link
IC REG LIN 3.3V 500MA SOT-223-3	Microchip Technology	1	$0.69	Link
IC REG LIN 2.8V 500MA SOT-223-3	Microchip Technology	1	$1.16	Link
IC REG LINEAR 1.5V 1A SOT-223-3	Microchip Technology	1	$1.61	Link
CONN FFC VERT 24POS 0.5MM SMD	GCT	1	$0.62	Link
CRYSTAL 32.7680KHZ 12.5PF SMD	Abracon LLC	1	$0.69	Link
Capacitor - 0.1µF / 50V (0805)	Yageo	12	(E-shop SMD)	Link
CAP CER 18PF 50V X8R 0805	KEMET	2	$0.20	Link
Capacitor - 0.001 µF / 50V (0805)	Samsung	3	(E-shop SMD)	Link

Capacitor - 10µF / 50V (0805)	muRata	6	(E-shop SMD)	Link
Connector - Micro USB-B	Amphenol FCi	1	(E-shop SMD)	Link
Diode - CDBA540-HF (DO214AC)	Comchip	2	(E-shop SMD)	Link
Diode - CDBA540-HF (DO214AC)	Comchip	3	(E-shop SMD)	Link
Resistor - 10KΩ 1%(1/8W) (0805)	Stackpole Electronics, Inc.	14	(E-shop SMD)	Link
RES 1M OHM 1% 1/8W 0805	TE Connectivity Passive Product	5	$0.50	Link
Resistor - 22Ω / 1% / (1/10W) (0603)	Yageo	1	(E-shop SMD)	Link
Switch - Tactile	Littelfuse	2	(E-shop SMD)	Link

Table 2. Parts List

4.2 Labor

The labor cost estimate assumes three team members working approximately 10 hours per week over 14 weeks at a rate of $40/hour, consistent with standard ECE 445 cost estimation guidelines.

Team Member	Hours	Rate ($/hr)	Total
Kyle Smith	140	$40	$5,600
Mithesh Balle	140	$40	$5,600
Nathan Hwang	140	$40	$5,600
Total	420	—	$16,800

Table 3. Estimated Labor Costs

The total estimated project cost, combining parts and labor, is approximately $16,950.

5. Conclusion

5.1 Accomplishments

The PutterIQ system was successfully designed, built, and demonstrated as a working sensor integrated putter that provides real time putting stroke analytics. The final prototype met all three high level requirements. The face angle measurement achieved an average error of approximately ±0.54°, within the ±0.6° target. Impact detection operated at 100% reliability during testing, and heel/center/toe classification reached 93% accuracy. The BLE communication subsystem delivered stroke packets with 100% transmission success, no duplicated or dropped strokes, and latency consistently below the 500 ms threshold. The physical integration added only 299 grams to the putter, well within the 400 gram limit, and 18 of 20 surveyed golfers rated the putter feel at 3 or higher on a 5 point scale. Beyond the core requirements, the project produced a fully functional mobile application built in Flutter with live stroke visualization and session history, capturing data on all stroke metrics. The camera subsystem demonstrated reliable ball detection and orientation line tracking with a 97% frame level detection rate and a 3.2% average drift angle error. The sensor fusion strategy, which evolved from treating the IMU and piezo sensors as independent channels to a fused architecture with the IMU as the primary timing anchor, proved to be a key design insight that significantly improved robustness across varying impact conditions.

The project advanced to the final round of the ECE 445 Spring 2026 competition and placed 2nd out of 103 teams.

5.2 Uncertainties

Several aspects of the system carry residual uncertainty. The face angle measurement, while meeting the ±0.6° target on average, relies on the BNO085's onboard sensor fusion, which is sensitive to magnetic interference. In environments with strong magnetic disturbances, such as near metal railings or electronic equipment, the heading accuracy may degrade beyond the measured performance. The system was verified on flat indoor putting greens, and behavior on outdoor surfaces with slope, wind, or variable lighting has not been fully characterized. The camera subsystem's ball detection pipeline was tuned for controlled indoor lighting and a white golf ball with a drawn equator line. Performance under direct sunlight, heavy shadow, or with balls of different colors has not been tested extensively. The low light adaptation mode improves robustness in dim conditions, but extreme low light remains a limitation. Impact location classification achieved 93% accuracy, but the confusion matrix showed that toe strikes were occasionally misclassified as heel strikes. This may be related to the physical placement of the piezo strips and how vibration propagates through the mallet head geometry. A larger test set across different putter models would be needed to determine if this is generalizable.

5.3 Ethical considerations

The PutterIQ system collects motion and performance data that is stored locally on the user's device. No data is transmitted to external servers or shared with third parties. In accordance with IEEE ethical guidelines regarding the protection of public welfare, the system is designed so that all collected data remains under the user's control and is used solely for personal performance analysis [IEEE Code of Ethics]. The system does not make claims about guaranteed improvement and presents its feedback transparently, clearly labeling computed metrics and their sources. The project also aligns with the ACM Code of Ethics in its honest representation of system capabilities. The mobile application clearly presents which metrics are measured directly (e.g., face angle, impact location) versus which are derived from post-processing (e.g., rotation rate, wobble). Limitations such as the camera's dependency on lighting conditions and the piezo sensors' reduced sensitivity on center strikes are documented rather than hidden. From a safety standpoint, the system operates at low voltages (3.3 V regulated from a 3.7 V Li-ion cell) and poses minimal electrical risk. The Bluetooth subsystem operates under FCC Part 15 regulations. Battery management follows standard Li-ion safety practices, and the mechanical integration does not compromise the structural integrity of the putter or introduce sharp edges or loose components during normal use.

5.4 Future work

Several directions exist for extending the PutterIQ platform. First, consolidating the two MCU board architecture into a single ESP32-S3 board would reduce weight, wiring complexity, and cost. This would require optimizing the firmware memory usage to allow Bluetooth and Wi-Fi to coexist or alternatively migrating to a dual core scheduling approach or physically add more SRAM.

Second, the camera processing pipeline currently runs on a local laptop. Porting a lightweight version of the ball detection and rotation analysis onto the ESP32-S3 itself, potentially using reduced-resolution frames and simplified detection heuristics, would eliminate the need for an external computer and make the system fully self-contained between the putter and the phone.

Third, the mobile application could be extended with more advanced analytics features, such as stroke-to-stroke consistency scoring, automatic drill recommendations based on detected weaknesses, and session comparison overlays. Integration with a cloud backend would enable cross-session trend analysis and multi-device synchronization.

Finally, the piezo-based impact location classification could be improved by adding a third sensor to provide better spatial resolution, or by exploring machine learning classifiers trained on a larger labeled dataset of heel, center, and toe impacts across different strike intensities and putter geometries.

References

[1] CEVA, “BNO08X Datasheet,” Rev. 1.17, 2023. https://www.ceva-ip.com/wp-content/uploads/2019/10/BNO080_085-Datasheet.pdf

[2] TE Connectivity, “Piezo Film Sensors Technical Manual,” https://www.te.com/usa-en/products/sensors/piezo-film-sensors.html

[3] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.

[4] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[5] IEEE, "IEEE Code of Ethics," Institute of Electrical and Electronics Engineers.

[6] ACM, "ACM Code of Ethics and Professional Conduct," Association for Computing Machinery.

[7] Federal Communications Commission, "Code of Federal Regulations Title 47, Part 15 — Radio Frequency Devices," FCC.

[8] Espressif Systems, *ESP32-S3-DEVKITC-1-N8R8*. Digi-Key Electronics. [Online]. Available: https://www.digikey.com/en/products/detail/espressif-systems/ESP32-S3-DEVKITC-1-N8R8/15295894

[9] Vishay Semiconductor Opto Division, *TSAL6400*. Digi-Key Electronics. [Online]. Available: https://www.digikey.com/en/products/detail/vishay-semiconductor-opto-division/TSAL6400/1681340

[10] International Electrotechnical Commission, "IEC 62133 — Safety requirements for portable sealed secondary cells and for batteries made from them, for use in portable applications," IEC.

[11] Seeed Technology Co., Ltd., *114993115*. Digi-Key Electronics. [Online]. Available: https://www.digikey.com/en/products/detail/seeed-technology-co-ltd/114993115/21277047

[13] Microchip Technology Inc., *TC1262-3.3VDBTR*. Digi-Key Electronics. [Online]. Available: https://www.digikey.com/en/products/detail/microchip-technology/TC1262-3-3VDBTR/443194

[14] Microchip Technology Inc., *TC1262-2.8VDB*. Digi-Key Electronics. [Online]. Available: https://www.digikey.com/en/products/detail/microchip-technology/TC1262-2-8VDB/2201984

[15] GCT (Global Connector Technology), *FFC2A32-24-T FFC/FPC Connector*. Digi-Key Electronics. [Online]. Available: https://www.digikey.com/en/products/detail/gct/FFC2A32-24-T/10656996

[16] Abracon LLC, *ABS06-32.768KHZ-1-T Crystal*. Digi-Key Electronics. [Online]. Available: https://www.digikey.com/en/products/detail/abracon-llc/ABS06-32-768KHZ-1-T/2217994

[17] Yageo Group, *X7R Multilayer Ceramic Capacitors Datasheet*. [Online]. Available: https://www.yageogroup.com/content/datasheet/asset/file/UPY-GPHC_X7R_6_3V-TO-250V

[18] KEMET, *C0805C180J5HACTU Ceramic Capacitor*. Digi-Key Electronics. [Online]. Available: https://www.digikey.com/en/products/detail/kemet/C0805C180J5HACTU/7948474

[19] Murata Manufacturing Co., Ltd., *GRM21BR61H106ME43 Datasheet*. [Online]. Available: https://search.murata.co.jp/Ceramy/image/img/A01X/G101/ENG/GRM21BR61H106ME43-01A.pdf

[20] Amphenol Communications Solutions, *10118194 Connector Drawing*. [Online]. Available: https://www.amphenol-cs.com/media/wysiwyg/files/drawing/10118194.pdf

[21] Diodes Incorporated, *CDBA540-HF / CDBA5200-HF Schottky Barrier Rectifier Datasheet*. [Online]. Available: https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/506/CDBA540-HF-CDBA5200-HF_RevC.pdf

[22] Stackpole Electronics Inc., *RMCF / RMCP Resistor Series Datasheet*. [Online]. Available: https://www.seielect.com/catalog/sei-rmcf_rmcp.pdf

[23] Yageo Corporation, *RC Series Chip Resistors Datasheet*. [Online]. Available: https://www.yageo.com/upload/media/product/products/datasheet/rchip/PYu-RC_Group_51_RoHS_L_12.pdf

[24] C&K Switches, *PTS645 Series Tactile Switch Datasheet*. [Online]. Available: https://www.ckswitches.com/media/1471/pts645.pdf

Appendix - Requirement & Verification Table

Table 4 System Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
1. High Level Requirement 1 a. ≤ ±0.6° face angle error b. ≤ 5 ms impact timing resolution		Y
2. High Level Requirement 2 a. ≥ 99% stroke packet transmission b. No Repeated strokes c. Max end-to-end latency of 500ms from impact		Y
3. High Level Requirement 3 a. < 400 grams added b. Putter does not restrict a natural putting stroke		Y
4. We need to ensure seamless connection between IMU and IC2	• Connect IMU to MCU via I2C. Run 100 continuous putting strokes. Log I2C NACK count. Verify error rate < 2% • Use I2C logic analyzer to confirm no bus errors during a full 30 minute putting session	Y
5. The IMU must measure linear acceleration along each dimension	• Record linear putt motion over a known displacement and time interval from video, estimate using derivation	Y
6. The IMU must measure angular velocity along each dimension	• Record angular putt motion over a known displacement an time interval video, estimate using derivation • Confirm values reported by IMU match reference within specification, and aim for +/- 500 degree/sec and resolution of 0.1 degree/sec	Y
7. IMU must accurately measure putter head velocity	• Record angular putt motion over a known displacement an time interval video, estimate using derivation • Aim for +/- 5% relative to measurements	Y
8. IMU must accurately measure face rotation while putting	• Compare system impact face angle against the observed outcome of repeated 10 ft putts and the	Y

	corresponding reference measurements from video - Confirm error ≤ ±0.6° across all test angles.	
9. Motion tracking operates throughout duration of the putt and without interruption	- Have a 30 minute putting session - Ensure that there are no gaps in data, or at least 98% of data is recorded live	Y
10. Ball Impact must be consistently recorded each stroke,.	- Take 300 putting strokes, must detect impact 98% of the time	Y
11. Ball Impact being able to classify it as either heel, toe, or center is extremely important for ball path analysis	- Observe at least 90% accuracy against a confusion matrix, so no more than 5 slight misreads per location	Y
12. Real time access to ball impact data	- Using an external timer and voltage readings, we will test 100 putts and ensure the average impact detection latency is ≤ 5 ms.	Y
13. We need to be able to detect mishits and errors Double hits may register only as the first or most recent putts	- We will again test 100 putts, and we must be able to detect the correct abnormality 95% of the time	Y
14. We want our sampling to occur with a minimum frequency of 2KHz during our impact	- We will configure the ESP32-S3 ADC for continuous sampling on both piezo channels - Verify sample timestamps show consistent ≤ 0.5 ms intervals using a timer	Y
15. Camera captures footage at a minimum of 20 fps.	- Configured frame counting program for burst captured - Verified average of 22.1fps > 20	Y
16. The camera must be able to tell the orientation of the ball in order to properly track movement.	- Run 100 burst captures - Verified orientation line was captured > 90 % of time between visible frames	Y
17. We want to be able to refine our ball rotation analysis to ensure our measured rotation rates are correct	- Set up protractor, recorded drift rates and cross referenced with actual drift from ideal path - Rotation rates should be within 5% of eachother	Y
18. The ball vision must be maintained throughout stroke and follow through.	- Perform 100 full strokes. Review captured video for illumination dropouts or occlusion - Ensure at least 98 putts are followed by camera module (98%)	Y
19. Stroke metrics must be computed as fast as possible so that we can transmit usable data directly to Bluetooth	- We must timestamp the impact detection itself as well as and final BLE packet transmission in firmware - We will log this for 100 strokes, and the mean time between these should be 500ms or less	Y

20. We need our MCU to transmit each reading from all of our measurement units	• We will perform a 100 stroke session with phone app logging received packets • We must verify that for each putt where we record any data that it is transmitted to the app, with a 100% MCU transfer success rate	Y
21. It is important that our local and transmitted data are aligned with no discrepancies	• Log stroke IDs on MCU and in app simultaneously for 50 strokes • Match each data point and ensure identical contents between app and raw data	Y
22. Our Bluetooth connection must remain stable while we are using the device	• We will initiate Bluetooth connection and run idle polling for 240 minutes at 10-foot range • We should have no disconnects	Y
23. In order for synchronization of our data transmission and calculations, our IMU and impact sensor data aligned within 5 ms.	• Simultaneously log IMU timestamp and piezo interrupt timestamp for 30 impacts • Compute time delta for each. Verify all within 5 ms.	Y
24. Our App needs to be snappy, and allow a golfer immediately after his putt to see what he did	• We will require that each putt reading is visible on the app within 2 seconds of the putt • Timestamp BLE packet reception and on screen render time in app logs for 30 strokes • Ensure average time is within 2 sec	Y
25. Our app has stroke data stored with 100% reliability and retrievability for in round trends	• We will perform a 100 stroke session. Close and reopen app, navigate to trend view, and verify all 100 strokes appear in history with correct data	Y
26. We need to be able to provide a stable 3.3V output at all times (under load or no load)	• We will connect regulator output to electronic load set to estimated max current draw. • We will measure output voltage with a multimeter. Verify 3.3 V ± 5% under no load and full load, from full battery (4.2 V) down to cutoff (3.5 V).	Y
27. We need our battery to be able to provide power for at least one full round of golf, ideally more	• We will charge our battery to full, and run the system with IMU streaming, BLE active, and camera at 20 fps. • We will do this 3 times, and record time until the system shuts down • The average time our system should run must be at least 4 hours	Y
28. We need to keep the weight of the putter close to the standard weight of a putter	• We will weigh the complete assembled putter (with all	Y

	electronics installed) and the baseline putter separately. - The difference must be less than 500 g, measured with a precision scale.	
29. It is paramount that our putter keeps a natural feel, and that golfers don't feel it is clunky to move	- Recruit 20 golfers (range of skill levels). Have each hit 10 putts with sensor mounted putter - Administer satisfaction survey (1–5 scale). Verify ≥ 18/20 rate 3 or higher for putter feel and balance	Y