PrecisionShot Training System

A Smart Laser Target System for Shot Accuracy Tracking and Real-Time Training Feedback

Senior Design I Final Report
EEL4914 - Senior Design I
Group 13

University of Central Florida
Department of Electrical and Computer Engineering

Anthony Fontana, CpE
DeLayne Russell, CpE
Kenn Pickavance, EE
Nicolas Koteff, EE

Reviewers: Dr. Wayesh Qarony, Dr. Jaesung Lee, Dr. Hadi Kamali, Dr. Kimia Azar, Dr. Saikat Dey, Dr. Shady Elashhab, Dr. Azadeh Vosoughi

Project Website: https://www.precisionshot.xyz/

Table of Contents

List of Figures and Tables

Chapter 1: Executive Summary

The PrecisionShot Training System is being developed as a portable dry-fire training platform that reduces the cost, travel, and facility requirements associated with live-ammunition practice. Existing laser-training products often provide only hit-or-miss feedback, depend on smartphone cameras, or require proprietary hardware or subscription software. PrecisionShot addresses these limitations by detecting laser impacts directly within the target, immediately displaying the shot location, and recording performance data without requiring a camera, cloud service, or continuous phone connection.

The system combines a standalone electronic target with a companion mobile application. Its sensing surface uses 500 phototransistors arranged in a 20-by-25 array across the active target area. Their analog outputs are routed through a multiplexer network to an ESP32-S3-WROOM-1-N8R8, which performs sensor scanning, ambient-light filtering, hit localization, scoring, training-mode control, and system management. Each sensing cell is paired with a surface-mount LED for immediate local feedback, while a 4.0-inch ST7796S capacitive touchscreen displays shot placement, score, mode, calibration, battery, and connection information. A commercial semi-solid-state USB-C power bank supplies the regulated 5 V rail, and a TLV62569DBVT buck regulator generates the 3.3 V rail for the controller, sensors, multiplexers, and storage hardware. A mechanically aligned grid faceplate provides optical isolation and helps block off-axis ambient light.

The software follows a device-first architecture. Embedded firmware is being written in C++ using ESP-IDF so the target can detect shots, calculate results, operate training modes, update the physical interface, and log events without a connected phone. The React Native and Expo mobile application communicates through Bluetooth Low Energy and provides remote control, live shot information, local training history, and performance analytics. Event-based JSON packets were selected for the first prototype because they are easy to inspect and debug. BLE was selected instead of Wi-Fi because the system does not require internet access, accounts, or cloud storage and benefits from low-power direct communication.

Senior Design I established the system architecture, measurable engineering specifications, component selections, hardware and software diagrams, PCB direction, enclosure concept, implementation schedule, and a prototype budget of approximately $922.20. Early firmware defines the touchscreen, microSD, and communication interfaces, and a BLE GATT server can transmit test-hit and score information while receiving basic commands. The mobile application interface and analytics workflow have also been prototyped. Initial phototransistor testing produced a stronger voltage response from a direct red laser than from bright white-light sources at short range, supporting threshold-based detection. Final accuracy, latency, reliability, wireless-range, and battery-runtime measurements have not yet been completed and remain validation goals rather than confirmed results.

The core performance targets require at least 90% of 30 controlled shots to be detected and reported within 2.0 cm of the actual impact location. At least 95% of detected shots must be processed and displayed on the target within 500 ms, while the mobile application must receive and display shot information within one second over BLE at distances up to 10 m. The system is also expected to operate for at least four continuous hours without an unexpected reset.

The principal risks are ambient-light interference, missed laser pulses during sequential scanning, multiplexer signal degradation, electrical noise from the display and LED power paths, and the mechanical alignment of all 500 sensors and LEDs with the faceplate openings. PCB revisions, enclosure reprints, component availability, and full-system integration also create schedule and budget risks.

Senior Design II will focus on sensor and multiplexer validation, final schematic and PCB completion, full-array fabrication, enclosure manufacturing, and hardware-software integration. The team will then calibrate the target under multiple indoor lighting conditions and measure shot-location accuracy, feedback latency, false detections, missed shots, app update delay, wireless behavior, voltage stability, and operating time. Reliable detection, scoring, local feedback, standalone controls, BLE communication, and a durable enclosure will be completed before lower-priority features such as automatic distance tracking, coaching feedback, and customizable drills.

Chapter 2: Project Description

2.1 Motivation and Background

For as long as firearms have been used in military, personal defense, hunting, and recreational applications, regular training has been essential to their safe and effective use. Firearm training traditionally uses live ammunition. Live ammunition, whether black powder and ball or modern cartridges, is expensive and requires access to a facility capable of safely stopping bullets while also managing the noise, smoke, and fumes produced by gunpowder. For these reasons, a firearm training system that does not require live ammunition or access to a shooting range would make firearm training cheaper and more accessible.

Our solution is to develop a comprehensive laser-based training system rather than only a standalone target. The user fires a laser at the target, which detects and displays where the laser hit. By pairing the target with a dedicated mobile application, the system will allow users to train, track performance statistics, review previous sessions, and improve their aim without needing to visit a shooting range.

In addition to reducing the cost of training, the system is intended to improve the quality of practice by providing immediate feedback after every shot. Traditional paper targets require shooters to stop their training session and walk downrange to inspect their results, disrupting the flow of practice and making it more difficult to recognize shooting patterns. A laser-based electronic target can instantly display shot placement, allowing users to make corrections in real time while maintaining a consistent training rhythm. By combining this immediate feedback with stored performance data, shooters can monitor long-term improvement, identify recurring errors, and objectively evaluate their progress across multiple training sessions.

Another important consideration is accessibility. Not all firearm owners have convenient access to a shooting range, and many indoor and outdoor ranges have limitations related to operating hours, travel distance, availability, or usage fees. A portable laser training system allows users to practice fundamental shooting skills, such as sight alignment, trigger control, and target transitions, in a safe environment without the logistical challenges associated with live-fire training. This makes regular practice more practical while reinforcing safe firearm-handling habits outside of a traditional shooting range.

2.2 Existing Products and Prior Related Work

Various laser training systems have been developed in the past. One example is the LaserLyte TLB-LMS Steel Tyme system [1]. This product uses a proprietary laser pistol and a target that lights up and produces a sound when hit. Another example is the Mantis BlackbeardX Dry Fire Laser Training System [2]. This system replaces the magazine and bolt of an AR-15 and connects to the user's phone to track where shots land on the target. The phone's camera records where each laser strike appears. Many other laser target systems simply indicate whether the target was hit or missed. In these systems, a laser detected anywhere on the target is registered as a hit without identifying the exact impact location.

Although these products demonstrate that laser-based firearm training is practical and effective, they also have several limitations. Many systems only indicate whether a target has been hit rather than determining the precise impact location. Others rely on a smartphone camera to detect laser strikes, making their performance dependent on camera placement, lighting conditions, and image-processing algorithms. Additionally, several commercial products require proprietary laser cartridges, dedicated training firearms, or subscription-based software, which can increase the overall cost and reduce compatibility with different firearm platforms.

The PrecisionShot Training System is designed to address many of these limitations by performing shot detection directly on the target using a dense array of phototransistors. Instead of depending on an external camera, the embedded hardware determines the location of each laser strike and immediately displays the result through an integrated LED array. This approach reduces external dependencies, improves response time, and allows the system to operate as a standalone device. The companion mobile application further expands the system by providing training history, performance statistics, configurable training modes, and wireless control without sacrificing standalone functionality.

2.3 Project Goals

The main goal of this project is to develop a target system that can detect and display the placement of a laser shot. Different settings and training modes will also be implemented so users have multiple ways to practice. The target should operate reliably under varying indoor lighting conditions and run from a rechargeable battery.

Table 2.1: PrecisionShot Project Goals and Success Criteria

Goal Type	Goal	Successful if
Basic	Develop a target that detects and locates laser shots	At least 90% of 30 controlled laser shots are detected and reported within 2.0 cm of the actual impact location under normal indoor lighting conditions.
Basic	Use a rechargeable battery to power the system	The system can be recharged through USB-C and operates for at least four continuous hours without an unexpected reset.
Basic	Provide immediate onboard feedback for each shot	The corresponding shot information is displayed on the target within 500 ms for at least 95% of detected shots.
Basic	Control target modes and settings	The buttons, touchscreen, and display allow the user to select modes, adjust

	using the onboard interface	calibration and distance settings, and reset the current session without a connected phone.
Basic	Develop a mobile application that connects to the target	The app connects through Bluetooth Low Energy and displays the correct shot location, score, mode, shot number, distance setting, and timestamp within one second for at least 95% of detected shots.
Advanced	Include user statistics and training history	At least 100 shot records remain available in the mobile application after it is closed and reopened.
Advanced	Allow the user to enter and record the shooting distance	The selected distance is shown on the target and mobile app and is stored with each shot record.
Advanced	Control target modes and settings through the mobile app	The app can change the mode, distance setting, calibration value, and current-session state while connected to the target.
Stretch	Track the angle of the training device to analyze user stability and form	The system measures and displays the shooting angle on the target or mobile application using additional sensing hardware.
Stretch	Develop additional training modes that simulate competitive shooting drills	The system supports advanced modes such as moving targets, randomly activated target areas, or timed drills.
Stretch	Include tips and educational material based on user performance	The application provides training suggestions based on patterns found in the user's stored shot data.
Stretch	Include customizable training modes	The user can create and save custom drills instead of relying only on preset training modes.

2.4 Project Objectives

The objectives of the PrecisionShot are as follows:

- Develop an array of phototransistors that detects incoming laser shots while rejecting normal ambient-light interference.
- Implement an LED array and onboard display to indicate where the laser shot was detected, with the onboard display updating within 500 ms for at least 95% of detected shots.
- Program a microcontroller to process phototransistor detection, LED indication, scoring, system settings, and training modes.
- Detect and report at least 90% of 30 controlled laser shots within 2.0 cm of the actual impact location under normal indoor lighting conditions.
- Ensure that the mobile application receives and displays the correct shot information within 1 second for at least 95% of detected shots over Bluetooth Low Energy at distances up to 10 m.

These objectives define the measurable performance goals for the PrecisionShot Training System. The system must not only detect a laser hit, but also determine the location of that hit with enough accuracy to provide useful training feedback. This requires the phototransistor array, LED feedback system, embedded software, and mobile application to work together as one complete system. The 500 ms onboard-display response requirement ensures that physical feedback appears quickly, while the 1 second app update requirement ensures that wireless data transmission remains fast enough for a smooth training experience.

Because the system is intended to function under varying indoor lighting conditions, the shot-detection process must be designed to separate valid laser strikes from unwanted ambient light. The microcontroller will be responsible for reading the sensor array, filtering the input signals, determining the shot location, updating the LED array and onboard display, and sending the shot data to the mobile application. These objectives provide the foundation for later hardware, software, and testing requirements.

2.5 Features and Functionalities

The PrecisionShot Training System combines embedded hardware with wireless software to provide a complete firearm training platform. While the target itself is capable of operating independently, integration with the companion mobile application provides additional functionality such as long-term performance tracking, configurable training modes, and remote control of system settings. Together, these features provide users with immediate visual feedback while collecting meaningful training data that can be used to evaluate accuracy and monitor improvement over time.

- The user will be able to change the settings of the target for different modes.
- The target will work wirelessly with an internal rechargeable battery.
- The system features full wireless app integration for logging training data, user statistics, and session history.
- The target will be able to operate under varying indoor lighting conditions.
- The target shows real-time shot placement on the physical target and on the app.

These features make the PrecisionShot system more than a simple hit-detection target. The physical target provides immediate local feedback through the LED array and onboard controls, while the mobile application expands the system into a training and performance-tracking tool. This allows users to review previous sessions, monitor accuracy trends, and adjust training settings without relying only on the physical target interface.

The combination of wireless operation, rechargeable power, real-time feedback, and mobile app integration supports the overall goal of creating a portable and accessible laser training system. Since the device can operate under different indoor lighting conditions and does not require a wired connection during normal use, it can be used in a variety of training environments. This flexibility helps make the system practical for repeated use while still supporting more advanced software features through the companion application.

2.6 Design Requirements

The design of the PrecisionShot Training System is driven by the need to provide fast, accurate, and reliable laser-shot detection under a variety of indoor operating conditions. The phototransistor array must be capable of detecting the selected laser wavelength while rejecting ambient light sources such as indoor lighting and reflections. The system must detect and report at least 90% of 30 controlled laser shots within 2.0 cm of the actual impact location under normal indoor lighting conditions. This requires both carefully selected sensor hardware and software-based threshold calibration to minimize false detections while maintaining high sensitivity to valid laser shots. The LED array must provide immediate visual feedback, while the onboard display must show the correct shot information within 500 ms for at least 95% of detected shots.

The embedded microcontroller must coordinate communication between all major hardware subsystems, including the phototransistor array, LED array, display, user controls, wireless communication system, and external storage. Processing must occur with sufficiently low latency to maintain real-time system performance while simultaneously supporting multiple training modes, user settings, and wireless communication with the companion mobile application. The mobile application must receive and display the correct shot information within 1 second for at least 95% of detected shots while connected through Bluetooth Low Energy. The firmware must also remain flexible enough to accommodate future software updates and additional training features.

Portability is another important design consideration. The complete system must operate from a rechargeable battery without requiring a wired power connection during normal use and must provide at least four continuous hours of operation without an unexpected reset. Bluetooth communication should allow the companion application to receive shot information, display user statistics, and remotely adjust target settings at distances up to 10 m without affecting the target's ability to function independently. Finally, the physical enclosure must protect the electronic components while remaining lightweight, durable, and suitable for repeated indoor use.

These design requirements connect the project objectives to the actual engineering choices that must be made during development. Sensor spacing, signal filtering, microcontroller selection, wireless communication, battery capacity, PCB routing, and enclosure design all affect whether the final system can meet its accuracy, speed, and portability goals. Each design choice must therefore support the larger requirement of building a system that is reliable, easy to use, and capable of providing meaningful training feedback in real time.

The system must also be designed with testing and future refinement in mind. Since the final prototype will include several interacting subsystems, the hardware and software should be organized so that individual components can be tested separately before full integration. This includes verifying sensor detection, LED response, wireless communication, display output, battery operation, and enclosure alignment. Designing for testability will reduce debugging time and improve the likelihood that the final prototype performs consistently during demonstration.

Chapter 3: Engineering Specifications

3.1 System-Level Engineering Specifications

3.1.1 Overall System Specifications

The PrecisionShot Training System is designed to provide a portable, standalone laser firearm training platform capable of accurately detecting laser impacts, providing immediate visual feedback, and recording training performance for later analysis. The system combines embedded hardware with a companion mobile application to create an affordable alternative to live-fire practice while maintaining realistic training conditions. Unlike many commercial laser training systems that depend on external cameras or internet connectivity, PrecisionShot performs all shot detection directly within the target, allowing it to operate independently without requiring additional equipment.

The system is intended primarily for indoor use with commercially available laser training cartridges or laser training devices that emit visible laser pulses. The sensing system must detect these laser pulses with sufficient sensitivity to accurately determine shot location while rejecting interference from normal ambient lighting. Because indoor lighting conditions may vary significantly between environments, the detection system must also support adjustable sensitivity through software calibration to maintain reliable performance under different operating conditions.

To successfully manage these environmental constraints, the physical design relies on a mechanical grid mesh faceplate integrated directly over the phototransistor matrix. Ambient indoor light, particularly overhead fluorescent or LED fixtures, can introduce significant background noise that risks saturating the optical sensors and masking short laser pulses. The geometric structure of the front-facing grid addresses this vulnerability by acting as a directional light filter, introducing physical isolation that restricts light capture exclusively to perpendicular, front-angled paths. By creating deep, angled cell walls that block oblique, off-axis ambient light from the sides, the faceplate establishes a controlled internal testing environment directly above the sensor lenses. This hardware isolation minimizes background interference before it reaches the electronics, lowering the baseline noise floor and allowing the firmware to consistently differentiate true laser impacts from static room lighting.

Real time operation is a fundamental requirement of the project. After a laser strike is detected, the system must process sensor information, estimate the impact location, calculate the appropriate score, and provide immediate visual feedback with minimal delay. Fast system response improves the training experience by allowing users to instantly identify shot placement and make corrective adjustments without interrupting practice.

The PrecisionShot target is designed to function as a complete standalone training device. Users should be able to operate the system directly from the target through an onboard display and physical controls without requiring a smartphone connection. Essential functions, including training mode selection, score display, distance selection, calibration adjustment, and battery monitoring, must remain available regardless of whether the companion application is connected.

Wireless communication provides additional functionality without becoming a dependency for normal operation. Bluetooth Low Energy communication allows the companion mobile application to receive shot information, store training history, display performance statistics, and remotely configure target settings. Since all primary processing occurs within the embedded hardware, temporary communication interruptions do not prevent the target from continuing normal operation.

Portability is another major design objective. The entire system must operate from an internal rechargeable battery while providing sufficient operating time for extended practice sessions. Power consumption must therefore be minimized through efficient hardware selection, event-driven communication, and intelligent firmware design. The physical enclosure must also protect internal electronics while remaining lightweight enough to be easily transported between training locations.

Finally, the system architecture must remain scalable for future improvements. The hardware and software should allow additional sensors, training modes, user profiles, communication features, and advanced analytics to be incorporated without requiring significant redesign of the overall platform. This modular approach extends the useful lifetime of the project while simplifying future development.

3.2 Component Specifications

3.2.1 Component Overview

The PrecisionShot Training System consists of several major hardware and software subsystems that work together to provide accurate shot detection, real-time processing, user feedback, and long-term performance analysis. Each subsystem has its own functional requirements that contribute to the overall performance and reliability of the system. The following sections describe the primary responsibilities and performance requirements of each subsystem without specifying the exact components selected for implementation.

3.2.2 Phototransistor Detection System

The sensing subsystem serves as the primary input mechanism for the PrecisionShot Training System. Its purpose is to detect incoming laser pulses across the target surface and convert the received optical energy into electrical signals that can be processed by the embedded controller. Since shot location accuracy depends directly on sensor performance, the detection system must provide consistent, repeatable measurements while minimizing false detections caused by ambient lighting.

The sensor array must provide sufficient spatial resolution to accurately estimate the location of each laser impact. Rather than simply detecting whether the target has been struck, the system must collect enough information to estimate the position of the laser strike with significantly greater precision than the spacing between individual sensors. The sensing subsystem must also respond quickly enough to detect short laser pulses generated by commercially available training cartridges.

The detection system must remain reliable under a variety of indoor lighting conditions. Software-adjustable detection thresholds should allow users to compensate for different lighting environments while maintaining sensitivity to valid laser impacts. The overall sensor architecture should also remain scalable so that future revisions of the project can increase sensing resolution without fundamentally changing the operating principles of the system.

3.2.3 Signal Multiplexing System

The sensing subsystem requires a method of reading a large number of analog sensor outputs using a limited number of microcontroller input channels. The multiplexing subsystem is responsible for sequentially routing sensor signals to the analog-to-digital converter while preserving the analog voltage information required for accurate shot location calculations.

The multiplexing architecture must support rapid scanning of the complete sensor array while introducing minimal signal distortion or electrical interference. Since sensor measurements are used to estimate laser position, the subsystem must preserve relative signal amplitudes instead of reducing each sensor output to a simple digital state. The design should also remain expandable so additional sensing channels can be incorporated without significantly increasing processing complexity or printed circuit board size.

3.2.4 Processing System

The processing subsystem coordinates the operation of every major component within the PrecisionShot Training System. It must continuously acquire sensor measurements, execute the shot detection algorithm, calculate impact location, determine scoring information, update user interfaces, and manage wireless communication while maintaining real-time system responsiveness.

The processor must provide sufficient computational performance to execute these tasks simultaneously without introducing noticeable delays during operation. It must also provide adequate memory, analog input capability, digital communication interfaces, and expansion resources to support both the current prototype and future feature development. The processing subsystem should remain capable of standalone operation even when external communication devices are unavailable.

3.2.5 Visual Feedback System

The visual feedback subsystem provides immediate confirmation of each detected laser impact. Once the processing subsystem determines the estimated shot location, the feedback system must clearly indicate the corresponding impact position on the target while simultaneously presenting important system information to the user.

The visual interface should provide sufficient brightness for indoor operation while maintaining low power consumption during battery-powered use. In addition to displaying shot location, the subsystem should support multiple colors or visual patterns for future training modes, system notifications, and diagnostic functions. Immediate visual response is essential for maintaining a realistic and uninterrupted training experience.

3.2.6 User Interface System

The user interface allows the PrecisionShot target to function independently of the companion mobile application. Physical controls and an onboard display provide users with direct access to essential operating functions, including training mode selection, calibration adjustment, score display, battery status, and system configuration.

The interface must remain intuitive and responsive while minimizing the number of user interactions required during normal operation. Users should be able to configure the target quickly without interrupting training sessions, and all essential settings should remain accessible even when wireless communication is unavailable.

3.2.7 Wireless Communication System

Wireless communication enables data exchange between the embedded target and the companion mobile application. The communication subsystem must reliably transmit processed shot information while also receiving user commands and configuration updates from the mobile application.

The communication protocol should prioritize low power consumption, reliable operation, and minimal latency. Since the target performs all critical processing locally, wireless communication should enhance the user experience without becoming necessary for basic functionality.

Temporary communication interruptions should not affect shot detection or normal target operation.

3.2.8 Power Management System

The power subsystem supplies stable operating voltages to every electronic component within the PrecisionShot Training System. Since the device is intended to operate from a rechargeable battery, power management must maximize operating time while maintaining stable electrical performance under varying load conditions.

Voltage regulation must provide clean, reliable power for both analog sensing circuitry and digital processing hardware. The subsystem should also monitor battery condition and provide users with battery status information to prevent unexpected shutdown during training sessions. Protection against overcharging, excessive discharge, and electrical faults should be incorporated to improve long-term reliability and user safety.

3.2.9 Companion Mobile Application

The companion mobile application extends the functionality of the PrecisionShot target by providing long-term data storage, performance analysis, remote configuration, and system diagnostics. Although the target is designed to function independently, the application provides users with a more comprehensive view of their training progress over time.

The application should maintain a local history of shot data, calculate performance metrics, and present graphical summaries that help users evaluate improvements in accuracy and consistency. It should also allow remote modification of target settings without requiring users to physically access the target during training. All application data should remain available without requiring cloud storage or internet connectivity, ensuring both user privacy and continued operation regardless of network availability.

3.3 Demonstrable Specifications

Table 3.1: Demonstrable Engineering Specifications and Verification Methods

Specification	Priority	Target Value	Measurement Method
Shot Detection and Localization Accuracy	Core	At least 90% of 30 controlled laser shots detected and reported within 2.0 cm of the actual impact location.	Fire 30 shots at predefined target locations and compare the reported coordinates with the marked coordinates.
Onboard Shot Processing and Display	Core	At least 95% of detected shots displayed within 500 ms with the correct location, score, mode, and shot number.	Measure the delay between detection and display update. Compare displayed values with the internal shot record.
Mobile Application Data Delivery and Display	Core	At least 95% of detected shots received and displayed by the mobile app within 1 second over BLE at distances up to 10 m.	Fire 30 shots while connected to the app. Compare the app's values and timestamps with the target's logs and display.
Ambient-Light Rejection	Medium	At least 90% detection reliability with no more than one false shot during a 10-minute idle test under dark, normal, and bright indoor lighting.	Repeat controlled shot and idle tests under three lighting conditions. Record missed detections and false positives.
Sensor Array Coverage and Mapping	Medium	100% of installed phototransistor channels must be readable and mapped to the correct physical position.	Activate each sensor region in diagnostic mode and compare the reported sensor ID with its physical location.

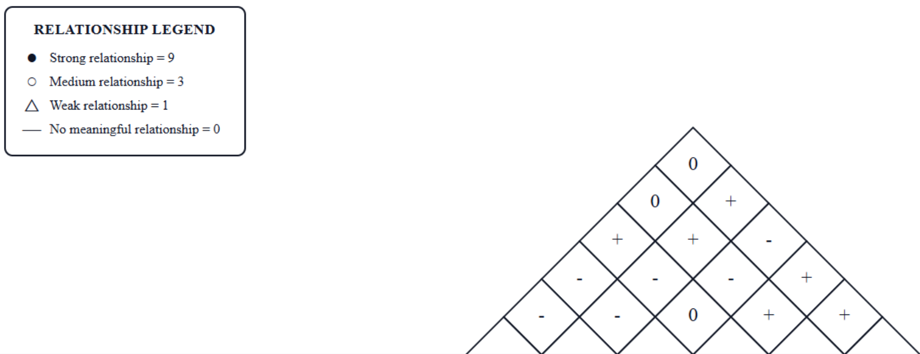
Standalone Target Operation	Medium	Shot detection, scoring, display feedback, mode selection, calibration, and reset must function without a connected phone.	Disconnect the app and complete a full training session using only the target controls and onboard display.
Local Shot-History Persistence	Medium	At least 100 consecutive shot records must remain available after the application is closed and reopened.	Generate or fire 100 shots, restart the app, and verify that every record remains stored with the correct values.
Stable Battery-Powered Operation	Medium	The system must operate for at least four continuous hours without resets, while the 5 V and 3.3 V rails remain within ±5% of their nominal values.	Fully charge the power source and operate the complete system for four hours. Measure both rails at regular intervals and monitor for resets or failures.

3.4 The House of Quality

The House of Quality was updated to reflect the full PrecisionShot system rather than only the laser target hardware. The customer requirements focus on the major user-facing goals of the project: accurate shot detection, fast feedback, ease of use, portability, lighting reliability, and wireless/app integration. These requirements were compared against the main engineering characteristics that drive the design, including sensor spacing, response time, power efficiency, ambient light filtering, wireless update delay, and user interface control.

The matrix shows that ambient light filtering and calibration carry the highest engineering importance. This is expected because the system depends on distinguishing valid laser strikes from normal indoor and outdoor lighting conditions. Shot detection response time and sensor spacing are also highly weighted because they directly affect real-time feedback and shot location accuracy. These priorities support the project's core objective of detecting laser impacts quickly and accurately while still allowing the target to function as a practical training device.

The correlation section highlights several design tradeoffs that will need to be managed during prototyping. Increasing sensor density improves shot accuracy, but it also increases scan complexity, routing difficulty, and potential power demand. More aggressive filtering improves reliability, but it must be implemented without slowing the target's response time. The competitor comparison also reinforces the design direction: PrecisionShot is intended to combine the immediate feedback of a physical target with the training history and remote-control benefits of an app-based system, while remaining portable and usable as a standalone device.



Customer requirements / Design direction & target	Customer importance rating (1 = low, 5 = high)	Percent of customer importance rating	Sensor spacing / array density	Local shot-detection response time	Battery runtime / power efficiency	Ambient-light false-trigger control	BLE app update delay	Standalone interface / training-mode control	Competitor research (1 = poor, 5 = excellent): LaserLyte Steel Tyme	Mantis BlackbeardX	PrecisionShot planned design
Design direction & target			▼ Center-zone sensor spacing no greater than 5 mm; outer-zone spacing no greater than 20 mm	▼ Laser detection and local LED feedback in less than 10 ms	▲ At least 10 hours of normal operation per charge	▼ Less than 1% false triggers during defined indoor and outdoor lighting tests	▼ New shot data displayed in the mobile app in less than 1 second	▲ At least three core training modes available through the onboard display and controls			
1. Accurate shot-location detection	5	20%	●	○	—	●	△	—	2	4	5
2. Fast real-time feedback	4	16%	△	●	△	△	○	△	5	3	5
3. Easy standalone setup and operation	4	16%	△	○	○	○	○	●	5	3	4
4. Portable and rechargeable	3	12%	△	△	●	—	△	○	4	4	4
5. Reliable across different lighting conditions	5	20%	○	△	—	●	—	△	4	2	4
6. Reliable BLE/mobile-app integration	4	16%	—	△	△	—	●	○	1	5	4
Importance rating			71	75	47	106	68	66			
Percent of importance			16%	17%	11%	25%	16%	15%			

Figure 3.2: House of Quality Matrix for the PrecisionShot Training System

3.5 Standards and Design Constraints Summary

Standards and design constraints are incorporated throughout the PrecisionShot hardware, software, and prototype-design chapters rather than being isolated in a separate chapter. This approach connects each requirement directly to the subsystem it affects. Laser safety, electrical protection, power-system safety, wireless communication, data privacy, manufacturability, ambient-light performance, mechanical alignment, cost, and schedule limitations are discussed alongside the related engineering decisions. Table 3.3 identifies the primary locations where these topics are addressed.

Table 3.3: Standards and Design Constraints

Area	Primary Standard or Design Constraint	Main Report Location
Laser and user safety	Laser classification, eye exposure, beam control, and warning requirements	Section 5.8.3
PCB and electrical design	Power distribution, grounding, test points, component protection, and PCB layout practices	Sections 5.6 and 5.7

Battery and power safety	Commercial power-bank protection, current monitoring, overcurrent protection, and voltage regulation	Sections 5.7.1–5.7.6
Wireless communication	BLE operation, connection reliability, communication protocols, and standalone functionality	Sections 6.4.2–6.4.5
Data privacy	Local application storage, no required cloud service, and single-device training records	Sections 6.3.4 and 6.4
Optical performance	Ambient-light rejection, sensor calibration, individual baselines, and optical grid isolation	Sections 5.2, 6.2.4, and 7.3
Mechanical constraints	Sensor alignment, enclosure dimensions, PCB mounting, component access, and structural protection	Chapter 7
Manufacturability	SMD assembly, PCB fabrication, stencil use, enclosure fabrication, and final assembly	Sections 5.2, 5.6, and 7.5
Economic constraints	Prototype cost, contingency planning, funding, and component availability	Chapter 8
Schedule constraints	Priority levels, subsystem dependencies, team responsibilities, and SD1 and SD2 milestones	Chapter 9

Together, these sections document how applicable standards and practical constraints influenced the final PrecisionShot design. The cross-reference is provided to make these considerations easier to identify without repeating the same technical material in a separate chapter.

Chapter 4: Research and Investigation

4.1 Existing Products and Shot-Detection Technologies

Before finalizing the design of PrecisionShot, existing commercial laser-based firearm training systems were researched to identify the capabilities, limitations, and design approaches currently used in the market. The products examined included the LaserHIT Basic Mini, Strikeman, Mantis Laser Academy, the VNSH Digital Steel Laser Target system, and the LaserLyte Trainer Target Steel Tyme Kit [88]-[92]. These products represent several different approaches to laser-based training. Some systems use a camera and software to locate the laser strike on a physical target, while others use dedicated electronic targets to detect laser illumination directly. Examining these products provided useful insight into the tradeoffs between cost, portability, feedback quality, system complexity, and the ability to determine shot location.

One of the most common approaches currently used in commercial laser-training systems is to place a laser cartridge inside the user's firearm and use a camera-equipped mobile device to observe the target. The LaserHIT Basic Mini is an example of this approach. The system uses a laser training cartridge that emits a short laser pulse when the firearm's firing pin or striker contacts the cartridge. A smartphone running the LaserHIT application is positioned toward a physical target and uses the camera to identify the laser impact. The application then displays the detected shot and provides feedback to the user. The Basic Mini package includes a laser cartridge, phone tripod, target materials, and a carrying case, with the system designed to provide a portable training setup [89].

The LaserHIT system demonstrates the advantages of using computer vision rather than dedicated sensing hardware across the entire target. A single camera can observe a large target area, allowing the system to determine the approximate location of the laser impact without requiring hundreds of individual electronic sensors. This reduces the amount of hardware required at the target itself. The system also benefits from the processing capabilities of the smartphone, which can perform image analysis and provide a graphical user interface without requiring a dedicated embedded processor in the target [89].

However, a camera-based system also introduces limitations. The performance of the system depends on the camera's ability to distinguish the laser from the surrounding environment. Ambient lighting, reflections, camera exposure, and the color and material of the target can affect detection performance. The camera must also be positioned correctly relative to the target. The LaserHIT system addresses some of these challenges through specially designed target materials and software-based processing. Nevertheless, the reliance on computer vision represents a fundamentally different design philosophy from PrecisionShot, which is designed around a dedicated physical phototransistor array for directly detecting the laser's position [89].

Strikeman uses a similar general architecture. Its standard system includes a laser cartridge, physical target, target frame and base, phone mount, tripod, and access to a mobile application. The laser cartridge produces a laser pulse when the firearm is fired, and the smartphone is used to observe and score the shot. The Strikeman application provides real-time shot scoring and progress tracking, with additional training features available through its Pro platform. The company also offers multiple system configurations, including standard and Marksman kits, as well as systems based on rechargeable laser hardware [88].

The primary advantage of the Strikeman approach is the integration of hardware and software into a complete training ecosystem. The user receives immediate feedback from the application, while training history and performance can be tracked over time. This provides a more complete training experience than a simple electronic target that only indicates whether a hit occurred. The software can also provide different drills and challenges, allowing the same hardware to be used for different training exercises [88].

The primary limitation of these camera-based systems is that the target itself does not directly measure the laser impact. Instead, the smartphone must observe the target and interpret the optical information. This makes the software and camera important parts of the sensing system. PrecisionShot takes a different approach by making the target itself an electronic sensing device. Instead of asking a camera to locate a laser spot within an image, PrecisionShot divides the target into a grid of phototransistors and directly measures the optical signal received by each sensor.

Mantis Laser Academy is another major example of a camera-based laser training system. The system uses a laser training cartridge, physical targets, a smartphone or tablet, and the Laser Academy application. When the firearm is fired, the cartridge emits a laser pulse that is detected by the camera of the mobile device. The application then records and scores the shot. Mantis provides multiple training modes, including single-target practice, multiple-target drills, timed exercises, and competitive modes. The system also provides a large library of training exercises and performance tracking features [90].

The Laser Academy system demonstrates an important trend in the current market: the value of a laser-training product is not limited to the physical sensor. Software has become a major part of the overall product. A single piece of hardware can support many different training modes when the application is responsible for scoring, timing, and user interaction. This provides significant flexibility and allows manufacturers to add features without substantially changing the physical target [90].

However, the same camera-based sensing approach remains subject to the limitations of optical image recognition. The mobile device must be able to observe the target and identify the laser pulse. As a result, factors such as lighting, target contrast, camera angle, and distance can affect the detection process. PrecisionShot's dedicated phototransistor array is intended to provide a more direct method of determining the location of the laser. Rather than processing a two-dimensional camera image, the system measures the analog light intensity detected at individual locations across the target surface.

A different approach is demonstrated by the VNSH Digital Steel Laser Target. Rather than using a camera to locate a laser impact on a target image, the system uses an electronic target that directly reacts when illuminated by the laser. According to the product description, the laser cartridge is used with a digital target that provides immediate feedback through a visual and audible response. The target can automatically reset after detecting a laser strike, allowing repeated practice without manually resetting the target [91].

The VNSH approach is advantageous because it reduces the dependence on a smartphone camera and software-based image processing. The target itself is responsible for detecting the laser and providing feedback. This can make the system simpler to operate and more immediate from the user's perspective. The product is also designed to be portable and rechargeable, allowing it to be used in different locations without requiring a permanent installation [91].

The primary limitation of this approach is that the electronic target is primarily designed to determine whether the laser has struck a target area rather than to provide a detailed two-dimensional measurement of the shot location. The target provides a reaction to a detected laser hit, but its design does not represent a high-resolution sensor array capable of determining the exact location of the laser across a large number of sensing points. PrecisionShot is intended to expand on this concept by using a large array of individually addressable phototransistors. This allows the system to preserve the analog intensity information from each sensor and use the combined sensor data to calculate the location of the laser impact.

The LaserLyte Trainer Target Steel Tyme Kit represents another electronic-target approach. The system includes two electronic targets and a laser-equipped training pistol. When the laser is directed at a target, the target provides visual feedback through LEDs and can also produce an audible sound designed to simulate the sound of a projectile striking steel. The system includes both a sound-enabled mode and a silent mode. The training pistol includes a resetting trigger and is designed to provide a physical training experience without using live ammunition [92].

The LaserLyte system emphasizes immediate feedback rather than detailed shot-location analysis. The user can quickly determine whether the laser was successfully directed at the electronic target based on the sound and visual response. This is useful for practicing basic sight alignment, trigger control, and rapid target engagement. The use of multiple independent targets also allows the shooter to practice transitioning between targets [92].

Compared with PrecisionShot, however, the LaserLyte system provides relatively limited information about the precise location of a laser strike. Its primary purpose is to determine whether the target has been successfully hit. PrecisionShot instead uses a dense phototransistor array to produce spatial information about the laser impact. The goal is not only to identify whether a laser was detected but also to determine where the laser struck and to use this information for shot-location analysis [92].

The commercial systems researched can therefore be divided into two major categories. The first category consists of camera-based systems, including LaserHIT, Strikeman, and Mantis Laser Academy. These systems use a smartphone or tablet camera to observe a physical target and software to identify and score laser impacts. Their primary strengths are flexibility, software-based training features, and the ability to use a large target area without installing a

sensor at every possible impact location. Their primary limitations include dependence on camera performance, environmental lighting, target visibility, and image-processing algorithms.
The second category consists of dedicated electronic targets, including the VNSH Digital Steel Laser Target and the LaserLyte Steel Tyme system. These systems directly detect laser illumination using electronic target hardware and provide immediate feedback through lights, sounds, or other target reactions. Their primary strengths are simplicity and immediate feedback. However, the amount of spatial information provided by these systems is more limited than the high-resolution location measurement intended by PrecisionShot.

The following table summarizes the major characteristics of the systems examined:

Table 4.1: Comparison of Existing Laser Training Products

Product	Detection Method	Feedback Method	Shot Location Information	Major Strength	Major Limitation
LaserHIT Basic Mini	Smartphone camera and computer vision	Mobile application	Camera-derived location	Portable and software-driven	Dependent on camera placement and lighting conditions
Strikeman	Smartphone camera and application	Real-time scoring and training application	Camera-derived location	Progress tracking and multiple training modes	Requires camera-based detection
Mantis Laser Academy	Smartphone or tablet camera and application	Scoring, drills, games, and performance tracking	Camera-derived location	Extensive software training ecosystem	Dependent on camera placement and optical conditions
VNSH Digital Steel Target	Dedicated electronic laser target	Audible and visual feedback	Target-area detection	Simple and immediate feedback	Limited spatial resolution
LaserLyte Steel Tyme	Dedicated electronic target	LEDs and simulated steel-impact sound	Target-hit detection	Immediate physical feedback	Primarily determines hit or miss

The research into these products directly influenced the design of PrecisionShot. Existing commercial systems demonstrate that laser-based firearm training can be implemented successfully using either camera-based computer vision or dedicated electronic targets. However, the research also revealed an opportunity for a system that combines the direct sensing approach of an electronic target with significantly greater spatial resolution. PrecisionShot addresses this opportunity through a large array of phototransistors distributed across the target surface.

The use of phototransistors provides an important distinction from simple digital target sensors. Each phototransistor produces an analog electrical response related to the amount of light it receives. Instead of treating each sensor as simply ON or OFF, the system measures the voltage associated with the detected light intensity. These analog readings are routed through analog multiplexers and measured by the ESP32's analog-to-digital converter. This allows the system to retain information about the relative strength of the optical signal at different sensor locations.
The use of a 500-sensor array also creates a direct relationship between the physical target and the electronic measurement system. The target surface is divided into a grid, with each phototransistor representing a specific region of the target. When a laser strikes the target, the sensors nearest the laser spot receive the greatest amount of light. By comparing the readings from the surrounding sensors, the system can estimate the location of the laser impact. This

provides a greater level of spatial information than a conventional electronic target that only reports whether a target area has been hit.

The sensor-array approach does introduce additional hardware complexity. A 500-sensor system requires substantial analog routing, multiplexing, power distribution, and signal processing. The ESP32 cannot directly dedicate an individual analog input to every phototransistor, so the sensor outputs must be multiplexed. The analog multiplexing architecture allows multiple sensor voltages to share the available ADC inputs while preserving the voltage information from each selected sensor. The ESP32 controls the multiplexer address lines, selects a sensor, measures its analog voltage, and then repeats the process for the remaining sensors.

This design represents a tradeoff between hardware complexity and measurement capability. A camera-based system can observe a large target using a single imaging sensor, but it requires image processing and is affected by the optical environment. A simple electronic target can provide immediate feedback with relatively little processing but may provide limited information about the exact shot location. PrecisionShot instead uses a dedicated optical sensor array to directly measure the spatial distribution of the laser signal. The resulting system requires more hardware than a simple electronic target but provides significantly more direct information about the location and intensity of the detected laser light.

The research also helped identify several design goals for the PrecisionShot system. First, the target should provide direct feedback without requiring the user to manually inspect the target after each shot. Second, the system should provide more spatial information than a simple hit/no-hit target. Third, the sensing system should be capable of preserving analog information rather than reducing every sensor to a binary state. Finally, the system should be capable of integrating the sensor data into a software interface that can provide meaningful feedback to the user.

Based on these requirements, PrecisionShot was designed as a hybrid system combining a physical optical sensor array with embedded processing. The phototransistor array provides the physical sensing mechanism, the analog multiplexers reduce the number of required ESP32 ADC inputs, and the ESP32 processes the sensor readings to determine the laser's location. This architecture differentiates PrecisionShot from the existing products studied during the research phase and establishes the design direction for the hardware and software developed in the remainder of the project.

4.2 Optical Sensors and Signal Filtering

Sections 5.2.2 and 5.2.3 establish the physical operating principles of a phototransistor and the specific part selected for the sensor array, the Everlight ALS-PT204-6C/L177, alongside two alternates, the Vishay TEPT5700 and TEPT4400, in Table 5.5. This section steps back one level and documents the broader technology-level research that supports that choice: why a phototransistor was selected over a photodiode or another sensing technology in the first place, and a fuller side-by-side comparison of the three candidate parts using the same criteria of spectral response, speed, sensitivity, package, cost, and availability.

4.2.1 Sensing Technology Comparison

A photodiode is the simpler of the two semiconductor options: it is a PN-junction device that converts incident light directly into a photocurrent and is normally operated in reverse bias, without needing the base-driven amplification a transistor provides. Because there is no internal current gain, a photodiode's output is small and linear with incident light, but its response is fast, typically limited only by junction capacitance, with usable bandwidth on the order of 0.35 divided by the response time. Photodiodes and phototransistors both convert incident light into electrical current, but photodiodes generally provide faster and more linear response, while phototransistors provide internal current gain and greater output sensitivity [62], [63]. A cadmium-sulfide photoresistor was also considered, but its comparatively slow rise and fall times make it poorly suited for detecting short laser pulses [64].

A phototransistor, as documented in Section 5.2.2, is built around the same photodiode junction but places it at the base of an NPN transistor so that the transistor's current gain amplifies the photocurrent into a much larger collector current. This is why phototransistors are consistently reported as more sensitive than bare photodiodes at the cost of a slower response: the same internal capacitance and carrier storage that produce the gain also limit how quickly the device can turn on and off, typically pushing response times into the low microsecond range rather than the nanosecond range a photodiode achieves. For PrecisionShot, the phototransistor's slower response is not a limiting factor, detecting a laser dry-fire pulse only requires microsecond-to-millisecond response times, well within phototransistor territory. The gain is also directly useful, since it lets each of the 500 sensor cells produce a usable voltage swing through a simple resistor-divider bias circuit instead of requiring a dedicated transimpedance amplifier per channel, which would have made the 500-channel array considerably more expensive and harder to route.

Three other sensing approaches were considered and set aside before narrowing to the phototransistor. The simplest alternative, a cadmium-sulfide photoresistor (LDR), was eliminated on speed alone: typical CdS LDR rise times run from roughly 10 ms up to several hundred milliseconds depending on illumination, with fall times that can exceed a full second, which would blur or miss the short optical pulse produced by a dry-fire laser cartridge entirely.

A position-sensitive detector was also considered because PSD devices can report the continuous centroid location of a light spot with high speed and resolution [65], [66]. A PSD is well suited to tracking a single moving beam, but PrecisionShot needs 500 independent hit/no-hit zones rather than one continuous coordinate, and PSDs are generally justified in high-volume applications specifically because they replace an imaging sensor's processing overhead for that single-spot case, using one PSD per sensor cell would be a costly way to answer what is fundamentally a yes/no question at each of 500 locations.

The last alternative, a camera-based approach, is exactly what several of the commercial systems reviewed in Section 4.1 (LaserHIT, Strikeman, Mantis Laser Academy) use instead of a dedicated sensor array. A single imaging sensor can observe a large target area without populating it with individual detectors, but that convenience comes at the cost of dependence on lighting, camera placement, and image-processing software rather than a direct electrical measurement at the point of impact. A published comparison of low-cost laser-spot measurement methods examined CMOS imaging, photodiode sensing, and photoresistive sensing, illustrating the additional hardware and processing requirements associated with camera-based detection [67].

Table 4.2: Optical Sensing Technology Comparison

Technology	Spectral Coverage	Speed	Sensitivity	Fit for 500-Cell Array
Photoresistor (LDR)	Visible only	Very slow (10s–100s ms)	Low, no gain	Eliminated, too slow for a laser pulse
Photodiode	Broad (UV–NIR)	Fastest (ns–µs)	Low output, needs an amplifier per channel	Possible, but adds a TIA per cell
Phototransistor	Broad (UV–NIR)	Fast (µs)	High, internal gain drives a resistor divider directly	Selected, Section 5.2.2 / Table 5.5
Position-sensitive detector	Broad	Fast, continuous position	High, but reports one coordinate, not many zones	Eliminated, wrong problem shape, higher cost per zone
Camera / CMOS imager	Broad	Frame-rate limited	High, but software-dependent	Eliminated, used by competing products in Section 4.1, dependent on lighting and image processing

4.2.2 Candidate Phototransistor Comparison and Final Selection Rationale

Once the phototransistor was selected as the sensing technology, Table 5.5 compared three real, sourceable parts, the Everlight ALS-PT204-6C/L177, the Vishay TEPT5700, and the Vishay TEPT4400, by size, spectral peak, and general reasoning. Table 4.3 extends that comparison with the datasheet values for spectral response, response speed, sensitivity, package, per-unit cost, and distributor availability, since the shot-detection front end depends on all of these together, not just spectral peak alone.

Table 4.3: Candidate Phototransistor Comparison for Red-Laser Detection

Part	Spectral Response	Rise / Fall Time	Sensitivity	Package	Cost (ea.)
ALS-PT204-6C/L177	Peak 630 nm, range 370–670 nm, ±30° half-angle	tr 0.11 ms / tf 0.12 ms (VCC = 5 V, RL = 7.5 kΩ)	15–150 µA at 100 lx (bin-dependent), dark current ≤ 0.1 µA	3 mm T-1 through-hole	~\$0.42–\$0.45
TEPT5700	Peak 570 nm, range 440–800 nm, ±50° half-angle	Not specified on datasheet	~75 µA typ. at 100 lx, dark current 3–50 nA	5 mm T-1¾ through-hole	~\$0.85
TEPT4400	Peak 570 nm, range 440–800 nm, ±30° half-angle	Not specified on datasheet	15–28 µA typ. at 20 lx (bin A), dark current 3 nA	3 mm T-1 through-hole	~\$0.15–\$0.85

Cost and general spectral placement were already covered in Table 5.5's reasoning column, so the additions in Table 4.3 were used to support the final decision. The most consequential difference is response time: the ALS-PT204-6C/L177 datasheet specifies a typical rise time of approximately 0.11 ms and a typical fall time of approximately 0.12 ms under its stated test conditions [68]. Neither the TEPT5700 nor the TEPT4400 datasheet specifies a response time at all, which is consistent with both parts being marketed specifically for slow, human-eye-matched backlight-dimming applications rather than pulse detection. Since PrecisionShot's shot-detection logic in Section 6.2.3 depends on catching a short laser pulse rather than a slowly changing ambient-light level, having a documented sub-millisecond response was a meaningful point in the ALS-PT204-6C/L177's favor even before the spectral-peak argument in Table 5.5 is considered.

The half-sensitivity viewing angle is the second differentiator, and it connects directly to the mechanical design in Section 7.3. The ALS-PT204-6C/L177 and TEPT4400 share a narrower ±30° half-angle, while the TEPT5700 is rated at a much wider ±50°. Because every sensor sits at the base of a 3 mm deep chamfered optical well behind the front grid plate, a narrower acceptance angle is already better matched to the mechanical light-blocking geometry described in Section 7.3, the grid cells are doing part of the same job the sensor's own acceptance angle does, and pairing a naturally wide-angle part with a narrow physical aperture wastes some of the sensor's usable sensitivity on angles the enclosure already blocks.

On cost, populating all 500 sensor positions with the ALS-PT204-6C/L177 at its Digi-Key unit price of roughly $0.42–$0.45 works out to approximately $210–$225, slightly above the $175 currently budgeted in Table 8.1 for 500 units at an assumed $0.35 each; this is worth flagging for the next budget revision, though it remains a small fraction of the overall $925 prototype estimate. All three parts are carried by major distributors (Digi-Key, Mouser) in sufficient quantity for a 500-unit build, so availability was not a differentiating factor in the final decision. For the production PCB, Section 5.2.4 identifies the ALS-PT19-315C/L177/TR8 as the SMD equivalent with a matching spectral response, keeping the through-hole and surface-mount parts of the design on the same underlying sensor characteristics.

Taken together, the technology-level comparison in Section 4.2.1 and the part-level comparison in this section support the selection already made in Section 5.2.3: a phototransistor is the right sensing technology for a 500-cell hit-detection array, and the ALS-PT204-6C/L177 is the strongest of the three candidate parts once response time and viewing-angle fit are considered alongside spectral peak and cost.

4.2.3 Analog Front-End and Filtering Approaches

The sensor array later described in Section 5.2.1 depends on more than the phototransistor itself. Before a laser strike can be compared against neighboring sensors and turned into a shot location by the weighted centroid method in Section 6.2.3, the raw photocurrent produced by each phototransistor has to be converted into a value the multiplexer, ADC, and firmware can act on. Five candidate front-end approaches were researched for this stage: a passive resistor-load conversion, a transimpedance amplifier, a comparator-based threshold stage, a dedicated analog filter placed ahead of the ADC, and firmware-side digital filtering applied after the ADC. These approaches are not mutually exclusive, and several can be combined in the same signal chain. Each is evaluated below on its signal conversion method, effective gain, noise behavior, threshold handling, added delay, and saturation behavior, using the ALS-PT204-6C/L177 phototransistor already selected for bench testing in Section 5.2.3 as the reference sensor for the calculations.

Resistor-Load Conversion - This is the approach already used for the bench testing described in Section 5.2.3 and Section 5.2.4, and it remains the simplest of the five. A single resistor connects the phototransistor's collector to the supply rail or ground, and the sensor's own manufacturer application note gives the governing relationship directly: Vout = Iph x RL, with the constraint that Vout(max) should stay at or below Vcc minus 0.4 V to avoid saturating the output. On the system's 3.3 V rail, that ceiling works out to roughly 2.9 V. The ALS-PT204-6C/L177 datasheet lists a dark current (ICEO) of 0.1 uA maximum at zero illuminance, a typical light current of 100 to 200 uA at 100 lux, and a typical light current of 1 to 2 mA at 1000 lux, all measured at a 5 V supply. Using a 1 kOhm load resistor as a representative value for a 3.3 V system, a 1000 lux reading of roughly 1.5 mA produces about 1.5 V, and a 100 lux reading of roughly 150 uA produces about 150 mV, both comfortably inside the available headroom. The effective gain of this stage is simply the load resistor value itself, 1 kOhm, or about 1 mV of output per uA of photocurrent.

The load resistor's own thermal noise can be estimated with the standard Johnson noise formula, en = sqrt(4kTR). At room temperature (T = 298 K) and RL = 1 kOhm, this works out to about 4.06 nV per root Hz. Using the phototransistor's own 0.11 ms typical rise time to estimate an equivalent bandwidth of roughly 0.35 divided by the rise time, or about 3.2 kHz, the integrated thermal noise over that bandwidth comes to roughly 0.23 uV rms, several orders of magnitude below even the smallest signal step considered above. This confirms that the load resistor's own thermal noise is not the limiting noise source in this design; ambient-light shot noise and dark-current drift, both already discussed qualitatively in Section 5.2.2 and Section 5.2.3, are the more significant contributors in practice.

Because this stage is entirely passive, threshold detection has to happen downstream, in the digital comparison and clustering logic described in Section 6.2.3. The delay through this stage is dominated by the phototransistor's own response time rather than by the resistor itself: the datasheet's 0.11 ms rise time and 0.22 ms fall time, measured at a 7.5 kOhm load, together with the multiplexer settling time of tens of nanoseconds and the ADC conversion time of a few microseconds to roughly 100 microseconds from Section 5.3.3, put the total per-channel latency somewhere in the range of 150 to 300 microseconds, with the sensor itself as the largest single contributor. A smaller load resistor than the datasheet's 7.5 kOhm test condition trims the RC-driven portion of that response somewhat, at the direct cost of lower gain, the same tradeoff the manufacturer's own Vout = Iph x RL relationship makes explicit. Saturation is bounded on two sides: the phototransistor's own collector-emitter saturation voltage (0.4 V maximum at 2 mA) and the resistor-driven output ceiling described above. Because a focused laser strike concentrates far more optical power on a single sensor than the diffuse test illuminance used in the datasheet's light-current figures, the center sensor under a direct hit should be expected to saturate rather than report a value proportional to the true peak intensity, which is consistent with the weighted-centroid method's reliance on the surrounding, unsaturated sensors for sub-cell position resolution rather than on the center reading alone.

Transimpedance Amplifier - A transimpedance amplifier, or TIA, replaces the passive load resistor with an operational amplifier configured so that its inverting input sits at a virtual ground and the phototransistor's photocurrent is forced through a feedback resistor, Rf, producing Vout = Iph x Rf [69]–[71]. Because the amplifier holds its input node near a fixed voltage rather than letting it swing with the load resistor, it does not modulate the phototransistor's own operating point the way the resistor-load approach does. The tradeoff for that benefit is bandwidth: the stage's cutoff frequency is approximately 1 divided by 2 pi times Rf times the total capacitance at the feedback node, Cf, which includes the phototransistor's own junction capacitance, the amplifier's input capacitance, and PCB stray capacitance. Using a representative Rf of 100 kOhm and an estimated Cf of 15 pF, the cutoff works out to roughly 106 kHz, or a time constant near 1.5 microseconds, about two orders of magnitude faster than the phototransistor's own 110 microsecond rise time. At this gain, the amplifier's own bandwidth would not be the limiting factor in the signal chain; the sensor's internal response would still dominate, the same conclusion reached for the resistor-load approach.

The noise advantage of a TIA over a simple resistor load is that its output noise scales only with the square root of Rf, while its gain scales linearly with Rf, so raising the feedback resistance to improve sensitivity at low light levels improves the signal-to-noise ratio rather than degrading it, an advantage worth considering if bench testing shows the resistor-load approach lacks sufficient margin at the low end of the ambient-light range. Output saturation with a TIA is bounded by the amplifier's own rail-to-rail output limits rather than by the phototransistor's collector-emitter voltage, which gives a more predictable and adjustable ceiling, though a fast, high-amplitude pulse can still be distorted rather than cleanly clipped if it exceeds the amplifier's slew rate. Threshold detection, as with the resistor-load approach, is typically still resolved downstream in firmware unless paired with a comparator stage. The practical obstacle to a TIA in this design is scale: giving each of the 500 phototransistors its own dedicated op-amp channel would add a meaningful amount of board area, per-unit cost, and 3.3 V rail current draw well beyond what the current component budget in Section 8.2 accounts for. A TIA only becomes practical here if placed once, or a small number of times, after the multiplexer network described in Section 5.3.1, in the same position the resistor-load's output currently feeds into the ADC, rather than ahead of each individual sensor.

Comparator-Based Threshold Detection - A comparator converts the analog sensor voltage directly into a binary logic output by comparing it against a fixed or adjustable reference voltage, effectively performing threshold detection in hardware rather than in firmware. Used alone with a single fixed reference, a signal hovering near that reference point produces rapid, repeated output transitions in the presence of even small amounts of noise. Adding hysteresis, implemented with a resistor network feeding back a fraction of the comparator's own output, separates the trip point into an upper threshold, VUT, and a lower threshold, VLT, with the difference between them, VHYST, defining a noise-immune band the input has to cross before the output changes state again. A comparator using only one fixed threshold may repeatedly change state when a noisy input remains near that threshold. Positive-feedback hysteresis creates separate rising and falling trip points, forming a noise-immune region that prevents unstable output transitions [72], [73]. Given that the resistor-load thermal noise calculated above comes to roughly 0.23 uV rms, even a modest hysteresis band on the order of tens of millivolts would reject that noise source many times over, meaning hysteresis sizing in this application would really be driven by ambient-light flicker and multiplexer channel-switching transients rather than by the phototransistor's own thermal noise floor. Comparator propagation delay is typically on the order of tens of nanoseconds, far faster than the phototransistor's own response time, so it would not add meaningfully to the total per-channel latency computed above.

The central limitation of a comparator for this application is that it discards amplitude information. A binary hit or no-hit output tells the firmware which sensor crossed the threshold, but not by how much, which conflicts directly with the weighted-centroid shot-location method in Section 6.2.3 that depends on comparing the relative strength of a triggered sensor against its neighbors to estimate a sub-cell impact position. Like the transimpedance amplifier, a dedicated comparator per sensor is impractical at 500 positions. The more realistic role for a comparator in this design would be as a shared, post-multiplexer hardware trigger, for example driving an interrupt pin on the ESP32-S3 the moment any scanned channel exceeds a coarse threshold, which could wake the firmware from a lower-power polling state without replacing the analog amplitude measurement the shot-location calculation actually depends on.

Dedicated Analog Filtering - A resistor-capacitor low-pass filter, or a higher-order active filter built from the same building blocks, can be placed between the sensor's output and the multiplexer or ADC input to attenuate high-frequency noise before it is digitized. The cutoff frequency of a simple single-pole RC filter is given by fc = 1 divided by 2 pi times R times C. Using representative values of R = 1 kOhm and C = 100 nF, the cutoff comes out to roughly 1.6 kHz, corresponding to a time constant of about 100 microseconds. Compared against the phototransistor's own equivalent bandwidth of roughly 3.2 kHz, derived earlier from its rise time, a filter cutoff set at or below that point necessarily slows the measured response further, and the added 100 microsecond time constant is comparable in size to the sensor's own intrinsic delay, meaning a per-channel analog filter at this cutoff could roughly double the total analog-domain latency depending on exactly where the cutoff is placed relative to the sensor's own bandwidth.

The benefit of this approach is that it attenuates high-frequency interference, including multiplexer channel-switching transients and the roughly 1.5 MHz switching noise from the TLV62569DBVT buck regulator described in Section 5.6.4, before that noise has a chance to alias into a sampled reading. As with the transimpedance amplifier and comparator, a discrete filter stage per sensor is not practical at 500 positions. The more realistic placement is a single shared filter stage after the multiplexer network and before the ADC input, filtering whichever channel is currently selected rather than filtering all 500 channels individually. Because a passive RC stage has unity gain in its passband, it does not change the saturation ceiling established by whichever conversion stage, resistor-load or transimpedance amplifier, sits ahead of it.

Firmware-Side Digital Filtering - Firmware-side filtering can apply an N-sample moving average after ADC conversion, reducing random sample-to-sample variation at the cost of added delay and reduced responsiveness to very short pulses [74], [75]. Under the standard assumption of uncorrelated, or white, sample-to-sample noise, averaging N samples reduces the noise amplitude by a factor of the square root of N. An 8-sample average, for example, reduces noise amplitude to roughly 1 divided by the square root of 8, or about 35 percent of its unfiltered value. This comes at the cost of a group delay of (N minus 1) divided by 2 samples. Using the roughly 9 microsecond per-sample figure from the hand-optimized ADC driver scenario in Table 5.11, an

8-tap average adds a delay of (8 minus 1) divided by 2, times 9 microseconds, or about 32 microseconds, small relative to both the phototransistor's own 110 to 220 microsecond response time and the roughly 100 microsecond time constant a comparable-cutoff analog filter would add per channel.

Because this filtering happens entirely in firmware, it adds no additional analog components, board area, or per-unit part cost, unlike the previous three approaches; the only cost is a small circular sample buffer per actively scanned channel and a handful of extra MCU cycles per reading, consistent with the microcontroller responsibilities already described in Section 6.2.1. The main risk is choosing too long an averaging window. Because the fastest laser-training cartridges researched in Section 5.8.2 can produce pulses as short as roughly 1 millisecond, in the LaserHIT HD variant, an averaging window that is long relative to that pulse duration would smooth the peak of a genuine laser strike toward the surrounding noise floor rather than simply rejecting noise around it, so the tap count should be chosen with the shortest expected pulse width in mind rather than for maximum noise rejection alone. Threshold detection with this approach is naturally handled downstream by the same digital threshold and clustering logic already described in Section 6.2.3, making digital filtering a complement to that logic rather than a competing design choice.

Table 4.4: Analog Front-End and Filtering Approach Comparison

Approach	Signal Conversion	Effective Gain	Noise Behavior	Threshold	Added Delay	Saturation Behavior
Resistor-Load Conversion	Vout = Iph x RL (passive)	RL itself; about 1 mV per uA at RL = 1 kOhm	About 0.23 uV rms thermal noise at RL = 1 kOhm; negligible next to the signal	Digital only, resolved post-ADC in firmware	About 150 to 300 us, dominated by the phototransistor's own response	Bounded by VCE(sat) (0.4 V max) and by Vout at or below Vcc minus 0.4 V
Transimpedance Amplifier	Vout = Iph x Rf (active, virtual ground)	Rf, set independent of sensor loading	Scales with the square root of Rf; better SNR than a resistor load at high gain	Digital, post-ADC, unless paired with a comparator stage	About 1.5 us at Rf = 100 kOhm, Cf = 15 pF; sensor response still dominates	Bounded by op-amp rail limits; slew-rate distortion possible on fast pulses
Comparator-Based Threshold Detection	Analog voltage vs. reference, binary output	Very high open-loop gain, hard limiting	Hysteresis band (VUT minus VLT) rejects noise below that width	Hardware trip point, fixed or hysteretic	Tens of ns propagation delay	Discards amplitude data; no proportional saturation behavior
Dedicated Analog Filtering	RC or active low-pass ahead of the ADC	Unity passband gain	Attenuates high-frequency noise (MUX transients, regulator switching)	None; filtering only, no threshold function	About 100 us added at fc = 1.6 kHz (R = 1 kOhm, C = 100 nF)	Unchanged; set by whichever conversion stage sits ahead of it

Firmware-Side Digital Filtering	N-sample moving average applied after the ADC	Unity DC gain	Noise reduced by the square root of N; about 65% reduction at N = 8	Digital threshold applied to the filtered value	(N-1)/2 samples; about 32 us at N = 8	Unchanged; risk of smoothing short pulses if N is too large

Considered together, these five approaches are not competing alternatives so much as layers of the same signal chain, and the calculations above point toward keeping the resistor-load conversion already investigated during phototransistor bench testing in Section 5.2.2 as the sensor-level stage, since it requires no active component per channel and its own thermal noise contribution was shown above to be negligible next to the signal levels involved. Firmware-side digital filtering pairs naturally with that choice, since it adds meaningful noise rejection at effectively no hardware cost and, at the sample rates achievable with an optimized ADC driver from Table 5.11, adds less delay than a comparable-cutoff analog filter would. A shared, post-multiplexer transimpedance amplifier or analog filter stage remains worth prototyping during the Senior Design 2 accuracy and latency testing milestone in Table 9.2 if bench results show the resistor-load path alone cannot deliver adequate signal-to-noise ratio at the low end of the detection range, but neither is recommended as a per-sensor addition given the resulting increase in board area, part count, and power draw across 500 positions. A per-channel comparator is not recommended for the primary detection path at all, since it would discard the relative amplitude information the weighted-centroid shot-location calculation in Section 6.2.3 depends on; if used, its more appropriate role is as a coarse hardware wake trigger rather than a replacement for the analog measurement itself.

4.3 Multiplexing, ADC, and Sensor Scanning

The large PrecisionShot sensor array requires a multiplexing system because the ESP32-S3 does not provide enough analog inputs to read every phototransistor directly. The team compared single-stage multiplexing, cascaded multiplexing, and external ADC architectures according to channel capacity, scan time, signal integrity, GPIO usage, cost, and implementation complexity.

A cascaded CD74HC4067 architecture was selected as the baseline approach because it can consolidate the full sensor array into a small number of ESP32-S3 ADC inputs while remaining significantly less expensive than high-channel-count external ADC systems. The additional switching delay introduced by the cascaded multiplexers is expected to be small compared with the ADC conversion time. Firmware will therefore use timer-controlled scanning and an optimized ADC read path to maintain the required response rate.

External ADC devices such as the MAX11128 and ADS7953 remain possible alternatives if prototype testing shows that the internal ADC cannot provide sufficient speed, stability, or signal quality. Final validation will measure complete-array scan time, channel settling, noise, and missed-pulse behavior under realistic operating conditions.

4.4 Microcontroller and LED Feedback Systems

The microcontroller research compared the ESP32 family, STM32 devices, and the Raspberry Pi RP2040 platform. Selection criteria included processing speed, memory, analog-input support, communication interfaces, wireless capability, power consumption, development tools, and cost. The ESP32-S3-WROOM-1-N8R8 was selected because it combines a dual-core processor, expanded flash and PSRAM, Bluetooth Low Energy, USB debugging, and the peripheral interfaces required for the display, storage, sensor scanning, and user controls.

The visual-feedback system requires one compact indicator for each active sensing position. Through-hole and surface-mount LEDs were compared based on size, assembly time, alignment, brightness, power consumption, and unit cost. Surface-mount LEDs were selected because they can be placed accurately using automated assembly and fit more easily within the dense target PCB.

The Inolux IN-S126ATB blue LED was selected for the current design because of its compact 1206 package, visibility, low unit cost, and compatibility with the planned PCB layout. The LEDs will be controlled through a multiplexed driver architecture so that immediate shot feedback can be provided without operating the entire array continuously or exceeding the system power budget.

4.4.1 MCU Technologies and Parts

Selecting a microcontroller for the system required comparing the major microcontroller families that are realistically available to a senior design team, both in terms of raw capability and in terms of cost, power behavior, and software support. Three families were evaluated side by side: the Espressif ESP32 line, the STMicroelectronics STM32 line, and the Raspberry Pi RP2040. Each family was scored against the same set of criteria: processor speed, on-chip memory, analog-to-digital conversion, general-purpose input and output pin count, Bluetooth Low Energy support, Wi-Fi support, power consumption, unit price, and the maturity of the available development tools. To keep the comparison grounded in real hardware rather than in datasheet abstractions, three specific modules were examined in depth: the Espressif ESP32-WROOM-32, the STMicroelectronics STM32F103C8T6 as sold on the low-cost "Blue Pill" development board, and the Raspberry Pi Pico W.

The ESP32-WROOM-32 family uses a dual-core Xtensa LX6 processor operating at up to 240 MHz and provides integrated Wi-Fi, Bluetooth, ADC channels, programmable GPIO, and several embedded communication interfaces [76]–[79]. Typical modules include approximately 520 KB of internal SRAM and external SPI flash, making the platform suitable for connected embedded-control applications. The chip includes 520 KB of internal SRAM and is typically paired with 4 MB of external SPI flash, although variants with up to 16 MB of flash exist. On the analog side, the ESP32 offers two SAR analog-to-digital converters spread across as many as 18 channels at 12-bit resolution, along with two digital-to-analog output pins. A typical module such as the ESP32-WROOM-32 breaks out 34 programmable GPIO pins that can be configured for PWM, SPI, I2C, I2S, UART, and capacitive touch sensing. Wireless connectivity is the ESP32's signature feature. It integrates 802.11 b/g/n Wi-Fi capable of roughly 150 Mbps alongside Bluetooth 4.2, supporting both classic Bluetooth and Bluetooth Low Energy on the same chip. Power draw scales with activity: active Wi-Fi use can pull upward of 80 to 160 mA, while deep sleep mode drops consumption to roughly 5 to 10 microamps, which makes the ESP32 well suited to battery-powered designs that wake only periodically. A bare ESP32-WROOM-32 module costs roughly 3 to 6 dollars, and the platform is supported by the Arduino IDE, the native ESP-IDF framework, PlatformIO, and MicroPython, giving it one of the most accessible tool ecosystems of any MCU on the market.

The STM32F103C8T6 used on common Blue Pill development boards contains a 32-bit ARM Cortex-M3 processor operating at up to 72 MHz, along with integrated ADCs, timers, communication peripherals, and general-purpose I/O [80]–[83]. Unlike the ESP32-S3, the standard Blue Pill board does not provide integrated Wi-Fi or Bluetooth and would require an additional wireless module for PrecisionShot. It carries 64 KB of flash and 20 KB of SRAM, noticeably less memory than either the ESP32 or the RP2040, which reflects the STM32 family's general design philosophy of trading memory headroom for low cost and tight power control. The chip provides two 12-bit ADCs across up to 10 channels and exposes roughly 37 GPIO pins, many of which are 5-volt tolerant, a convenience the ESP32 and RP2040 do not offer. Unlike the ESP32 and the Pico W variant of the RP2040, the base STM32 family has no built-in Wi-Fi or Bluetooth radio, so wireless functionality has to be added through an external module such as an ESP8266 co-processor or a BLE radio connected over SPI or UART. Active current draw on the Blue Pill sits around 25 to 30 mA, and standby modes can bring that down into the microamp range. Boards are inexpensive, with a Blue Pill module typically selling for 2 to 3 dollars, while higher-end STM32 Nucleo and Discovery boards run from about 10 to 25 dollars depending on the core. Development is handled through STM32CubeIDE and STM32CubeMX, ST's own free toolchain, as well as Keil MDK and an Arduino-compatible core for teams that prefer that workflow.

The Raspberry Pi RP2040 uses a dual-core ARM Cortex-M0+ processor operating at up to 133 MHz, includes 264 KB of SRAM, and provides programmable I/O state machines for implementing specialized digital interfaces [84]–[86]. Wireless communication is available on the Pico W through an additional radio device, while low-power behavior varies significantly with firmware configuration and may not match the deepest sleep performance available from the ESP32-S3 [87]. Instead the RP2040 connects to an external QSPI flash chip, typically 2 MB on Raspberry Pi's own Pico boards, with support for up to 16 MB. The chip includes a single ADC unit exposing 3 usable 12-bit channels capable of 500 thousand samples per second, and it breaks out 26 multi-function GPIO pins, 23 of which serve as general digital I/O once the ADC-capable pins are set aside. A standout feature is the set of 8 programmable I/O, or PIO, state machines, which let the RP2040 bit-bang custom high-speed protocols such as addressable LED timing entirely in hardware, something neither the ESP32 nor the STM32 can do natively. The base RP2040 has no radio at all. Wireless capability is only available on the Pico W and Pico WH variants, which add an Infineon CYW43439 chip supporting 802.11 b/g/n Wi-Fi and Bluetooth 5.2. Power behavior is less aggressive than the ESP32's: the datasheet lists a lowest-power dormant mode drawing around 180 microamps, though most real-world sleep implementations land closer to 1 to 2 mA, and active current with Wi-Fi transmitting can spike above 80 mA. A Raspberry Pi Pico costs about 4 dollars, and the Pico W adds roughly 2 dollars for the wireless chip, putting it in the same price bracket as the ESP32. The RP2040 is programmed in C or C++ through the official Raspberry Pi Pico SDK, or in MicroPython and CircuitPython, both of which are well documented and beginner friendly.

Table 4.5: MCU Part Comparison

Specification	ESP32-WROOM-32	STM32F103C8T6 (Blue Pill)	Raspberry Pi Pico W (RP2040)
Core / speed	Dual-core Xtensa LX6, 80-240 MHz	Single-core Cortex-M3, up to 72 MHz	Dual-core Cortex-M0+, up to 133 MHz
SRAM	520 KB	20 KB	264 KB
Flash	4 MB external (up to 16 MB on other variants)	64 KB internal	2 MB external (up to 16 MB supported)
ADC	2 x 12-bit SAR, up to 18 channels	2 x 12-bit, up to 10 channels	1 x 12-bit, 3 usable channels, 500 ksps
GPIO	34 programmable pins	About 37 pins, many 5V tolerant	26 pins, 23 usable as digital-only I/O
Bluetooth	Bluetooth 4.2, Classic and BLE	None on-chip, requires external module	Bluetooth 5.2 (Pico W only)
Wi-Fi	802.11 b/g/n, about 150 Mbps	None on-chip, requires external module	802.11 b/g/n via Infineon CYW43439
Active power	About 80-160 mA with Wi-Fi active	About 25-30 mA	Up to 80+ mA with Wi-Fi active
Sleep power	About 5-10 uA in deep sleep	Low microamp range in standby	About 180 uA minimum (datasheet), 1-2 mA typical
Unit price	About $3-6	About $2-3	About $6
Tooling	Arduino IDE, ESP-IDF, PlatformIO, MicroPython	STM32CubeIDE, STM32CubeMX, Keil MDK, Arduino core	Pico C/C++ SDK, MicroPython, CircuitPython

Each family favors a different point in the design space. The ESP32 offers the strongest combination of processing headroom, memory, and built-in wireless support, along with the lowest sleep current of the three, which makes it a strong default choice for any subsystem that needs both Wi-Fi and Bluetooth alongside moderate computation. The STM32 sacrifices onboard wireless and working memory in exchange for very low unit cost, broad 5-volt-tolerant I/O, and access to ST's mature, industrial-grade toolchain, making it a natural fit for a subsystem that only needs wired sensing or motor control and does not need to talk to a phone or a network directly. The RP2040 sits between the two: it has the most SRAM of the three, unmatched flexibility for custom digital protocols through its PIO state machines, and Wi-Fi and Bluetooth on the Pico W variant, but its sleep currents are noticeably higher than the ESP32's, which makes it a better fit where flexible I/O and raw core count matter more than long-term battery life. These trade-offs, together with the specific throughput, connectivity, and power budget required by each subsystem, guided the final part selection described later in this report.

4.5 Display, Storage, and Wireless Communication

4.5.1 Display Technology and Touchscreen Interface Selection

The original PrecisionShot design did not include a touchscreen. Early concepts considered using a small character LCD or scoreboard-style display that would only show basic information such as score, mode, and connection status. Under this approach, most settings would have been controlled through physical buttons or the mobile application, while Bluetooth communication would operate in the background.

Following additional research led by DeLayne Russell, the team found that touchscreen integration with the ESP32-S3 was more practical than originally expected. Modern display modules combine graphical output, touch control, and well-documented communication interfaces in a single device. This reduced the need for several physical buttons while allowing the user to navigate modes, calibration settings, score information, and system status directly from the target.

A touchscreen was also considered more user-friendly because users are already familiar with touch-based controls on phones, vehicles, appliances, and other electronic devices. The selected display is the Hosyond 4.0-inch 320 × 480 capacitive TFT touchscreen module, which uses the ST7796S display controller and communicates with the ESP32-S3 through SPI. Its FT6336U capacitive touch controller uses I2C, allowing the display and touch functions to operate without requiring an excessive number of GPIO pins.

The 4.0-inch display provides sufficient space for score, shot placement, mode selection, calibration, battery information, and Bluetooth status while remaining affordable and practical for the current enclosure. A future professional or second-generation version could use a larger, brighter, and more expensive touchscreen, but the selected module provides an appropriate balance of cost, size, usability, and integration difficulty for the prototype.

4.5.2 Removable Storage and microSD Card Selection

The final PrecisionShot prototype will include removable storage using a microSD card. The card slot will be positioned near the side of the enclosure so the user can remove the card without opening the entire device. Shot history, timestamps, scores, accuracy values, system settings, and debugging records can be stored locally and transferred to a laptop when needed. The exported data could then be reviewed directly or converted into a spreadsheet for additional analysis.

Because each shot record contains only a small amount of text-based information, the system does not require a high-capacity storage device. A low-capacity microSD card would provide more than enough space for the expected training history. The prototype will likely use a 16 GB card because one is readily available, even though the system is unlikely to use more than a small portion of that capacity.

More advanced storage standards, including SD Express, were considered unnecessary for this application. Their higher transfer speeds are intended for large files and high-bandwidth applications, while PrecisionShot will primarily store small JSON or text-based records. A standard microSD or microSDHC card therefore provides sufficient speed, capacity, availability, and compatibility at a lower cost.

MicroSD was selected over a full-size SD card because it requires less PCB and enclosure space while remaining familiar to most users. If testing shows that the smaller card is difficult to access or handle, a full-size SD card interface could be considered, but microSD remains the preferred option for the current prototype.

4.5.3 Wireless Technology Comparison and BLE Selection

The original wireless design used SvelteKit to create a browser-based control interface. Under this approach, the ESP32-S3 would host the generated webpage through a local web server, and the user would connect to the target through a local network address using a phone or computer browser. This would have avoided the need to install a dedicated application.

This approach was not selected after additional research and faculty feedback. A locally hosted web interface could behave differently depending on the user’s router, modem, firewall settings, device permissions, DHCP configuration, or local hostname support. These differences could make setup inconsistent and introduce connection problems that are outside the team’s control.

Bluetooth Low Energy was therefore selected as the primary wireless communication method. BLE allows the phone to connect directly to the ESP32-S3 without requiring internet access, a router, or changes to the user’s home network. It also provides lower power consumption than a continuously active Wi-Fi connection and is well suited for transmitting small event-based packets containing shot results, scores, settings, and system status. Further discussion of the mobile application implementation is provided in Section 4.6.2.

4.6 Firmware, Mobile Application, and Database Technologies

4.6.1 Firmware Development Frameworks

The PrecisionShot firmware must control sensor scanning, LED feedback, scoring, the touchscreen, Bluetooth communication, and local logging. Because several of these functions must operate at the same time, the selected framework needs to support reliable timing, direct hardware control, debugging, and an organized project structure.

Arduino was considered because it is simple to set up and has many existing libraries for ESP32 hardware. It is useful for quickly testing individual components, but its simplified structure becomes harder to manage when the system includes time-sensitive scanning, display updates, BLE communication, and storage operations.

PlatformIO provides stronger project organization, dependency management, and Visual Studio Code integration. However, it still depends on either Arduino or ESP-IDF as the underlying framework, adding another configuration layer without providing a major technical advantage for this project.

ESP-IDF was selected as the final firmware framework because it is the official development platform for the ESP32-S3 and provides direct access to GPIO, ADC, timers, SPI, I2C, BLE, and FreeRTOS. This allows sensor scanning and LED refreshing to run as controlled tasks while display updates, scoring, communication, and logging operate separately. Although ESP-IDF has a greater learning curve, it provides the control and debugging support needed for the final PrecisionShot firmware.

4.6.2 Mobile Application Framework and Communication Approach

The PrecisionShot mobile application is being developed using React Native, Expo, and TypeScript. React Native allows the interface and application logic to be developed from one shared codebase, while TypeScript improves code organization and helps identify errors during development. Expo provides project configuration, device testing, Fast Refresh, and simplified Android builds, allowing interface changes to be viewed quickly during development.

Development was also supported by generative-AI and code-assistance tools, including Google Gemini, OpenAI ChatGPT, Anthropic Claude, and GitHub Copilot, which were used for interface brainstorming, code review, debugging assistance, and comparison of possible implementation approaches [33]–[36]. CSS and Tailwind-style design references are widely available online, making it easier to compare layouts and develop a consistent visual theme. These tools support development, while the team remains responsible for reviewing, testing, and correcting the final application code.

The interface uses a modern enterprise-style appearance while including visual elements commonly associated with firearm and training applications. The selected color scheme is intended to remain professional, readable, and easy to navigate. Dark and light display modes are also being considered so users can select the appearance that is most comfortable for their environment. Future accessibility improvements may include color-blind display modes, larger interface elements, and voice or dictation support.

DeLayne Russell's experience in software engineering and more than seven years of UI/UX design helped simplify development of both the mobile interface and the touchscreen interface used on the target. Expo's development workflow, Fast Refresh, and Android build tools made early interface development relatively straightforward or, in less technical terms, a "piece of cake." Android Studio provides the Android SDK, Gradle tools, device logging, and native build support required as the application moves beyond standard Expo Go testing and begins deeper Bluetooth integration.

The current interface represents the first prototype version and has not yet completed formal user testing. The next stage will involve gathering feedback from friends and family before expanding testing to firearm enthusiasts and other potential users. Multiple interface revisions may be required to confirm that the application remains clear, accessible, and practical during an actual training session.

4.6.3 Database Technologies and Future Data Management

For the current PrecisionShot prototype, training data will be stored using JSON files instead of a traditional relational database. Each record only needs to contain a small amount of information, such as a timestamp, shot location, score, accuracy, distance, and active training mode. Because this data requires only a few kilobytes per user, a lightweight non-relational approach is sufficient for development, debugging, and demonstration. This may also remain the final prototype solution because it is simple to inspect, transfer, and troubleshoot.

SQLite remains the preferred option if the application later requires more organized local storage, faster searches, or larger training histories. It would allow shot records to be grouped by user, date, session, mode, and distance without requiring an internet connection or external server.

For a commercial consumer version, the application could remain entirely local and operate without accounts or cloud storage. A future enterprise version could use Firebase, Vercel, and SvelteKit to support secure cloud profiles, centralized session history, and organization-level analytics. Training agencies could create and manage multiple user profiles, connect several PrecisionShot units, and review performance using identifiers such as employee names, badge numbers, or email addresses. This would allow administrators to identify accuracy trends and determine when additional training may be beneficial. Any cloud-based version would require stronger authentication, privacy controls, and data-security protections than the standalone prototype.

4.7 Power Architecture, Consumption, and Runtime

4.7.1 Power Architecture and Voltage Regulation

PrecisionShot will use a 5000 mAh commercial power bank as its primary source. The power bank supplies a regulated 5 V rail for higher-power components such as the touchscreen and LED feedback system. A separate 3.3 V rail will power the ESP32-S3, multiplexers, phototransistor circuitry, microSD card, and supporting logic.

The 3.3 V rail will be generated using the TLV6256DBVT synchronous buck converter. It accepts the 5 V input and can provide up to 2 A output. This provides a stable shared voltage for the ESP32-S3 and CD74HC4067 multiplexers while keeping analog sensor signals within the ESP32 ADC range.

Decoupling capacitors will be placed near the ESP32-S3, multiplexers, display, and storage interface. Additional bulk capacitance will support the display and LED array during sudden changes in current. Sensitive phototransistor traces will also be separated from the switching regulator and LED-current paths where practical.

Ignoring every other system component and considering only the regulator's approximately 25 µA operating quiescent current, a 5000 mAh battery would theoretically last about 200,000 hours, or 22.8 years. This is only a mathematical comparison; battery self-discharge and the power bank's internal electronics would reduce the real result considerably.

4.7.2 System Power Consumption

The primary hardware loads in this section are the 500-phototransistor array, the CD74HC4067 multiplexer network, and the timer-multiplexed LED feedback array. The phototransistor and multiplexer circuitry should consume relatively little power during normal operation. The selected ALS-PT19 sensor produces approximately 15 µA at 100 lux, although its actual current will vary with lighting and the selected load resistor. The CD74HC4067 is a low-power CMOS device with quiescent current measured in microamps, so even several dozen multiplexers should remain a minor system load.

Earlier feedback-system research considered the Würth 150080VS75000 green LED and the Lumimax LEDDC-5CARGB RGB module as possible indicator devices [16]–[18]. The final design subsequently moved to the Inolux IN-S126ATB blue LED, so the final LED-array power estimate must be calculated from the Inolux specifications and verified through prototype measurements. Because the 20 × 25 matrix uses timer-based multiplexing, only one row will be active at a time. If all 25 LEDs in that row operate at 5 to 10 mA, the active array would draw approximately 125 to 250 mA before accounting for the driver circuitry. Actual consumption should be lower when only individual hits or partial target outlines are displayed.

A preliminary planning range of approximately 0.35 to 0.90 W will therefore be used for the sensors, multiplexers, LED array, and associated driver circuitry. This estimate will need to be replaced with direct current measurements after the 4 × 4 prototype and final LED-driver architecture are completed.

Ignoring the ESP32-S3, display, BLE, storage, and all other system loads, a 5000 mAh power bank containing approximately 18.5 Wh would theoretically operate these hardware subsystems for roughly 17 to 45 hours, assuming about 85% usable energy. A central estimate of approximately 25 hours will be used until prototype measurements are available.

4.7.3 Software, Display, BLE, and Storage Power Estimates

Software does not consume power separately from the hardware; instead, the firmware workload determines how long the ESP32-S3 remains active. Because PrecisionShot uses timer-based sensor scanning and LED control, the microcontroller will remain active during training rather than spending significant time in deep sleep. Bluetooth Low Energy is also built into the ESP32-S3, so its power use is included within the microcontroller estimate rather than treated as a separate component.

For preliminary calculations, the ESP32-S3 is estimated to average approximately 70 to 130 mA at 3.3 V, depending on processor activity, BLE traffic, display updates, and peripheral usage. A central planning estimate of 100 mA, or approximately 0.33 W, will be used. Espressif measurements show that current consumption varies considerably depending on processor workload, enabled peripherals, and whether power-management features can be used.

The selected 4.0-inch ST7796S touchscreen is expected to be the largest continuous load within this group. Its documentation lists a backlight current of approximately 103 mA and total power consumption of approximately 0.5 W. The backlight can therefore have a greater effect on runtime than BLE or storage, especially when operated continuously at full brightness.

The microSD card should contribute relatively little to average power consumption because shot records are small and writes occur only after events. A preliminary allowance of 3 to 10 mA at 3.3 V is used for intermittent logging. Power consumption can be reduced further by storing records temporarily in memory and writing several records to the card at once instead of performing constant individual writes.

The selected 5000 mAh power bank stores approximately 18.5 Wh at its nominal internal battery voltage. Assuming approximately 85% of that energy remains usable after conversion losses, the ESP32-S3, touchscreen, BLE connection, and microSD card would consume approximately 0.74 to 0.96 W. This produces a theoretical runtime of approximately 16 to 21 hours, with a central estimate of about 18.5 hours. This calculation excludes the phototransistor array, multiplexers, voltage-regulation losses beyond the assumed conversion efficiency, and the 500-LED feedback array.

Firmware power consumption can be reduced by using event-based BLE packets, avoiding unnecessary screen redraws, lowering the touchscreen backlight when idle, and buffering microSD writes. Continuously illuminated target-outline LEDs are not included in this estimate and may become one of the largest system loads depending on their brightness and duty cycle.

4.7.4 Estimated Battery Runtime

The following values are preliminary planning estimates intended for graphing and report development. They are not measured results and should be replaced with data collected from the completed prototype. The average operating condition assumes normal display brightness, active BLE communication, periodic microSD writes, and partial LED illumination rather than every LED operating continuously.

Table 4.6: Estimated Average Power Consumption by Subsystem

Subsystem	Estimated Power	Percentage of Total Power
ESP32-S3 processing and BLE	0.33 W	22.30%
TFT display and touch controller	0.50 W	33.80%
LED array and driver circuitry	0.45 W	30.40%
Phototransistor array	0.025 W	1.70%
Multiplexer network	0.015 W	1.00%
microSD card storage	0.025 W	1.70%
Buttons, indicators, and supporting logic	0.135 W	9.10%
Estimated total	1.48 W	100%

The display, LED feedback system, and ESP32-S3 are expected to account for most of the system's power consumption. Together, these three subsystems represent approximately 86.5% of the average estimated load. The sensor array, multiplexers, and microSD card consume significantly less power and should have a smaller effect on overall runtime.

For battery-runtime modeling, the 5000 mAh power bank is treated as having approximately 15.7 Wh of usable energy after estimated conversion losses. The best-case load is 1.09 W, the average load is 1.48 W, and the worst-case load is 1.86 W.

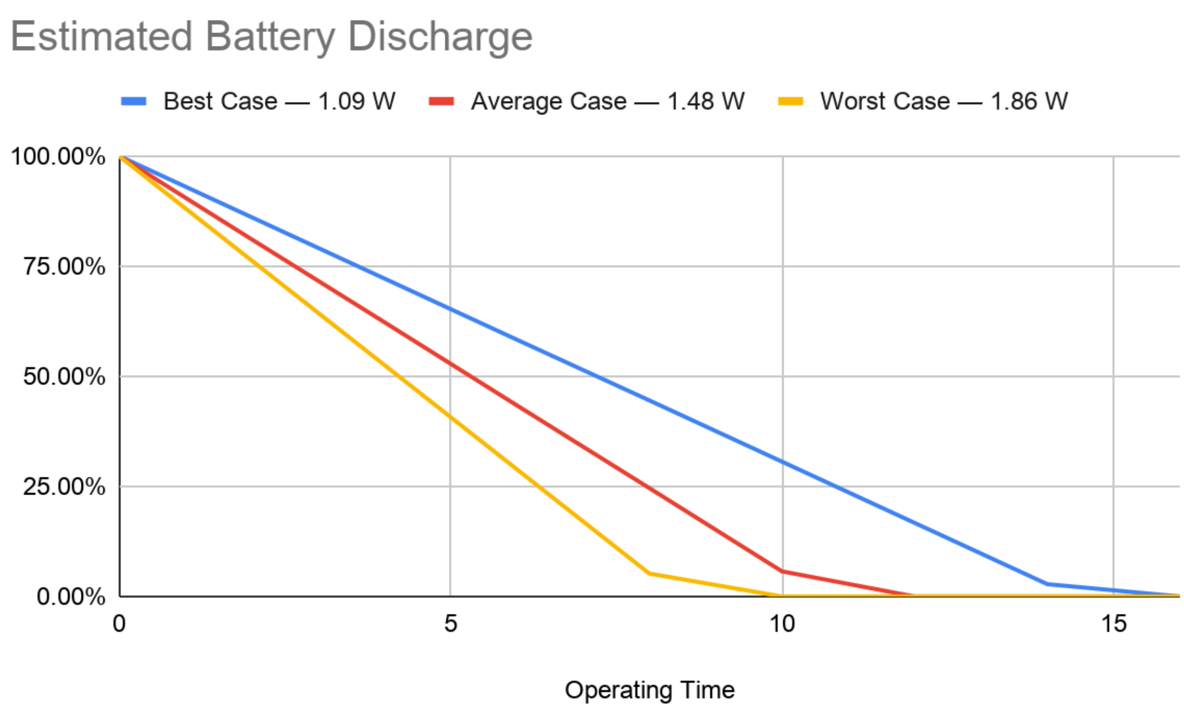


Figure 4.7: Estimated Battery Percentage Over Time

These values assume a simplified linear discharge rate. Actual battery percentage will not decrease perfectly linearly because power-bank efficiency, display brightness, LED activity, wireless traffic, and battery behavior will change during operation. Even under the preliminary worst-case estimate, the system should operate for approximately 8.4 hours, exceeding the project's four-hour minimum runtime requirement.

4.8 Enclosure, Optical Grid, and PCB Architecture

During the initial conceptual phase of the PrecisionShot design, the target matrix was configured as a 28-row by 20-column array encompassing 560 potential grid positions. In that preliminary architecture, only 400 nodes were designated as active phototransistor/LED pairs. The remaining 160 positions were intentionally unpopulated, creating intentional dead zones near the outer boundaries of the target frame. This concept was intended to simulate complete miss-fires when a shot landed beyond the primary scoring rings. However, empirical testing and system simulations revealed that unpopulated grid nodes compromised spatial tracking continuity, introduced edge-detection ambiguity, and artificially degraded overall system accuracy without offering meaningful diagnostic value to the user.

To resolve these tracking inconsistencies, the architecture required an operational revision: either fully populate all available grid locations or resize the mechanical frame to eliminate inactive nodes. Retaining the original 28x20 footprint and fully populating all 560 positions would have preserved the physical dimensions of standard firing-range paper targets. However, scaling to 560 active sensor/LED nodes introduced significant engineering and financial drawbacks:

Component and Assembly Costs: Adding 160 active nodes significantly increased the bill-of-materials (BOM) cost due to the additional phototransistors, surface-mount LEDs, debouncing components, and signal routing traces.

Multiplexer and Routing Complexity: Expanding to 560 active nodes exceeded the channel capacity of the planned analog multiplexer architecture. Supporting those additional signals would have necessitated extra multiplexer ICs, expanded control lines from the ESP32-S3 microcontroller, increased PCB trace density, and added hardware cost.

An alternative proposal considered scaling the matrix down to a square 20x20 grid (400 total nodes) to constrain component count and circuit complexity. While financially advantageous, a square 1:1 aspect ratio distorted the traditional silhouette of standard silhouette and bullseye targets, reducing the functional realism of the training experience.

To balance optical accuracy, physical proportions, and hardware constraints, a 25-row by 20-column matrix containing exactly 500 active nodes was selected as the final design. Operating on a uniform 0.5-inch grid pitch, this 12.5-inch by 10.0-inch active tracking layout achieves the following objectives:

100% Node Utilization: Eliminates inactive dead zones, ensuring continuous coordinate coverage across the entire active target faceplate.

Proportional Realism: Preserves a 5:4 aspect ratio that closely mimics the vertical geometry of standard firearm target silhouettes.

Controlled Hardware Overhead: Restricts the total active sensor count to 500 pairs, optimizing multiplexer channel allocation and maintaining micro-second scan loop iterations without inflating system power consumption or component expenditures.

Because the PrecisionShot Training System combines a 500-position phototransistor and LED sensor field with a compact control console inside a single physical enclosure (Section 7.2), the printed circuit board supporting these subsystems could not follow a typical senior design board layout. Before the specific layer stack, ground strategy, and component layout described in Section 5.6 could be finalized, the team first had to decide how the sensor field and control electronics would be physically organized onto printed circuit board hardware, and whether that hardware should take the form of one continuous board or several smaller, interconnected boards.

4.8.1 PCB Architecture Comparison

Integrating the 500-node optical matrix into a functional prototype required resolving physical envelope and optical isolation constraints. The total physical frame encompasses a 20.0-inch tall by 14.0-inch wide overall footprint, which is split into two primary functional areas: a lower 14.0-inch by 14.0-inch target housing containing the 12.5-inch by 10.0-inch active sensor matrix, and an upper 6.0-inch by 14.0-inch control console. This physical segregation isolates sensitive low-level analog signal paths from the digital noise generated by high-speed SPI/I2C communication lines and switching power components.

To optimize light capture while blocking ambient room glare, a custom front-facing grid faceplate was designed with an array of 500 individual 12.7 mm by 12.7 mm cells. Rather than using flat vertical walls, each cell features four inner side walls sloped at a 45-degree angle extending 4.1 mm down into the 6.1 mm thick plate, forming an inverted chamfered funnel. This geometry mechanically channels off-axis laser light down to a 4.5 mm by 4.5 mm aperture shelf containing a 2.5 mm window centered directly over each underlying ALS-PT204-6C/L177 phototransistor. Recessing the sensors behind these 6.1 mm deep mechanical tunnels establishes a restricted reception cone that blocks side-angle ambient light. Furthermore, to provide direct feedback illumination without blinding the underlying sensor, a dedicated 3.2 mm tunnel is angled into the upper chamfer wall to isolate each 3 mm blue visual feedback LED, preventing light leakage across adjacent nodes.

To prevent point-source LED hot-spotting and maintain stable laser detection, an optical filtering layer was integrated over the active grid area. Material evaluation compared a Frosted/Matte Clear Cast Acrylic (P95 finish) against a Translucent White / Opal Acrylic Sheet (Style 2447, 55% Transmittance). While the matte frosted clear sheet provided high forward transmission, it failed to fully eliminate harsh point-source LED glare. The 1/8-inch (3.0 mm) Translucent White 55% transmittance acrylic was selected because it successfully homogenizes the blue LED emission into uniform illuminated target zones while permitting sufficient laser energy to pass through to the phototransistors above firmware detection thresholds.

4.8.2 Control Console Interface and Internal Component Management

The upper 6.0-inch control section houses the user interface components and main processing hardware. The primary display consists of a 4.0-inch 320 x 480 TFT LCD driven by an ST7796S controller via SPI, paired with an I2C capacitive touch interface. To complement the touchscreen, three NKK EB2011-BG panel-mount, threaded pushbuttons (6.1 mm panel thickness compatible) are mounted directly adjacent to the LCD for hardware-level Mode Selection, Session Reset, and Ambient Calibration interrupts.

The internal power subsystem accommodates a heavy KU XIU semi-solid-state battery bank positioned in a reinforced rear compartment equipped with internal vertical ribs. To prevent parasitic battery drain during storage, a physical single-pole single-throw (SPST) rocker switch is placed inline with the primary 5V power trace on the main PCB immediately following the battery connection. A dedicated chassis opening exposes the battery bank's USB-C port to enable high-speed recharging without opening the main enclosure. To maintain thermal stability across the internal voltage regulators, microcontroller, and power bank during extended training sessions, 26 passive ventilation slits measuring 6.35 mm wide are integrated across the rear housing.

4.8.3 PCB Architecture Comparison and Fabrication Analysis

Two board-level architectures were compared: a single monolithic PCB spanning the full enclosure footprint, and a multi-board approach in which the sensor field would be built from several smaller tiled boards joined to a separate control board through board-to-board connectors or ribbon cables. The multi-board approach is common in large commercial LED signage and sensor arrays, where identical tiles can be manufactured, tested, and replaced independently. For PrecisionShot, however, this approach introduces a severe tolerance stack-up problem, since each of the 500 active sensing positions must align within sub-millimeter tolerances across its corresponding aperture in the 3D-printed target faceplate. Every seam between tiled boards becomes an additional source of positional error that firmware calibration cannot absorb. A tiled design would also require numerous board-to-board connectors solely to route shared analog references, multiplexer control lines, and ground planes between tiles, adding potential failure points and assembly steps without meaningfully reducing cost at a single prototype volume. For these reasons, a single continuous PCB was selected as the baseline board architecture.

Committing to a single board architecture at this scale was evaluated against real fabrication and assembly limits before being finalized. PrecisionShot's finished board occupies approximately 508 mm by 356 mm (20 in by 14 in), which fits within the standard panel limits listed by PCB fabrication services for single-piece prototype boards, allowing the bare board to be produced without sub-board splitting. However, automated assembly lines often restrict board sizes to roughly 250 mm by 500 mm. Turnkey automated assembly was therefore impractical, reinforcing the team's strategy of procuring bare boards and stencils, followed by in-house SMT component placement and reflow targeting the 500 sensor pairs and distributed multiplexer clusters.

Board thickness and layer count were evaluated under these same spatial constraints. A four-layer stack would provide dedicated internal power and ground planes, but the added copper and lamination cost scales directly with board area; at nearly 2 square feet of total surface area, a four-layer configuration would consume a disproportionate share of the hardware budget. A two-layer, 1.6 mm FR-4 board was chosen instead, incorporating a poured ground plane on the bottom layer and short, direct trace routing on the top layer for sensor, multiplexer, and control signals. The standard 1.6 mm thickness was retained specifically because the PCB must span the full vertical length of the chassis without bowing. A thinner board would flex under mechanical loading, shifting the phototransistor lenses out of alignment with the 4.5 mm by 4.5 mm aperture shelves and 2.5 mm alignment windows in the faceplate.

Component placement across the PCB mirrors the two-zone layout of the enclosure, placing the lower phototransistor/LED field and local multiplexer clusters away from the top control region housing the ESP32-S3, LCD interface, and buck regulation circuitry. Placing switching regulators far from the low-current phototransistor traces minimizes electromagnetic coupling into analog sensor lines. Finally, to prevent central sagging across the 20-inch board span, mounting points are organized as a distributed matrix of non-conductive nylon standoffs (M4 size, 40 mm length) securing the PCB flat against internal enclosure bosses, ensuring precise mechanical alignment across all 500 active tracking nodes.

Chapter 5: Hardware Design

5.1 System Hardware Architecture

5.1.1 System Hardware Overview

The hardware for the PrecisionShot Training System is organized into six major subsystems that work together to detect, localize, and display the result of each laser shot in real time. This organization separates the system along functional lines rather than physical board boundaries, which keeps the sensing, processing, feedback, interface, and power subsystems easy to test independently before they are integrated onto the final PCB.

1. A phototransistor array for shot detection, consisting of 500 active sensing positions distributed across five concentric target zones (Section 5.2).
2. An analog multiplexer and readout network that consolidates the 500-sensor array down to a small number of ESP32-S3 ADC channels (Section 5.3).
3. An Espressif ESP32-S3-WROOM-1-N8R8 control module that reads the sensor and multiplexer network, drives the display and LED array, manages local storage, and handles wireless communication (Section 5.4) [3], [4].
4. A surface-mount LED hit-indicator array, with one LED paired to each active sensor cell to provide immediate local feedback (Section 5.2.5).
5. A 4.0-inch ST7796S capacitive TFT touchscreen and physical push buttons for standalone user interaction (Section 5.5).
6. A power delivery system built around a commercial KUXIU semi-solid-state USB-C power bank and a TLV62569DBVT buck regulator, along with a microSD card for local event logging (Section 5.7).

Each subsystem depends on at least one of the others to function. The sensor array cannot report a shot location without the multiplexer network to route its signals, the ESP32-S3 cannot calculate or display a result without both of those upstream stages, and none of the digital hardware can operate without a stable regulated power rail. The remainder of this chapter documents each subsystem individually before describing how they connect on the shared PCB.

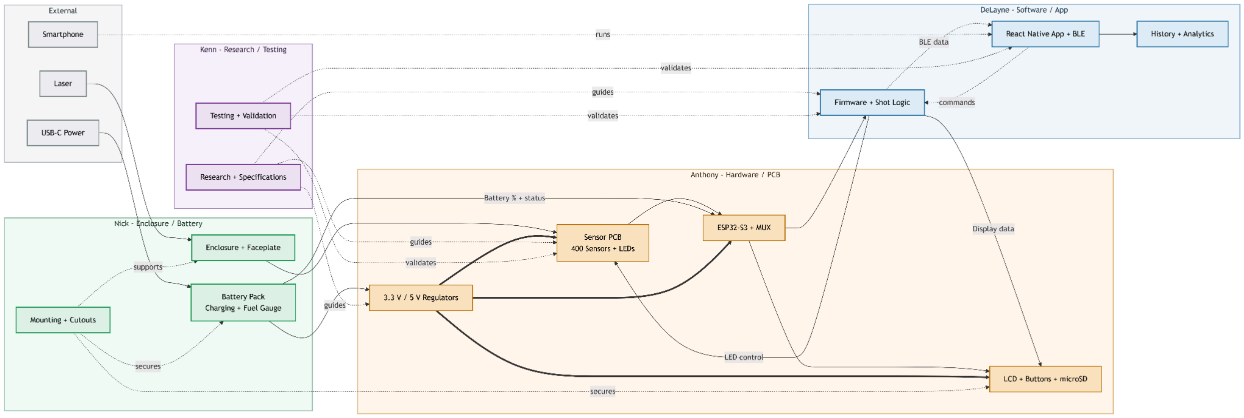


Figure 5.1: Hardware Block Diagram and Team Responsibility Breakdown

5.1.2 Hardware Block Diagram

The PrecisionShot hardware path begins with the power system. The commercial KUXIU semi-solid-state power bank supplies power over a standard USB-C connection and functions as the system's regulated 5 V rail. A TLV62569DBVT synchronous buck converter steps this 5 V input down to the 3.3 V rail required by the digital logic (Section 5.7.4). The 3.3 V rail powers the ESP32-S3-WROOM-1-N8R8, microSD card, phototransistor array, and multiplexer network, while the 5 V rail powers the TFT display and LED array directly.

From here, focus shifts to the MCU, which is the primary way each part of the hardware system interfaces with the others. The ESP32-S3-WROOM-1-N8R8 supports Wi-Fi and Bluetooth Low Energy, with BLE selected for direct, low-power communication between the target and companion mobile application [19](Section 5.4.2). It communicates with the surrounding hardware over its GPIO pins, which interface with the TFT display, microSD card, and physical push buttons. These same GPIO pins also serve as ADC inputs, which are necessary for reading the analog voltage produced by the multiplexer network as it scans the phototransistor array.

This block-level path, power distribution first and MCU-centered signal routing second, mirrors how the system is physically laid out on the PCB, with the sensor array and its supporting analog circuitry occupying the lower half of the board and the control and interface hardware occupying the upper half (Section 5.6.1). Figure 5.2 traces this same signal path in more detail, showing how a single laser strike moves from the phototransistor array through the multiplexer network to the ESP32-S3's ADC.

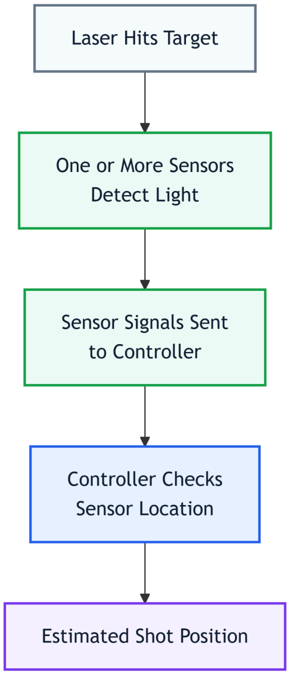


Figure 5.2: Sensor Detection Signal Flow

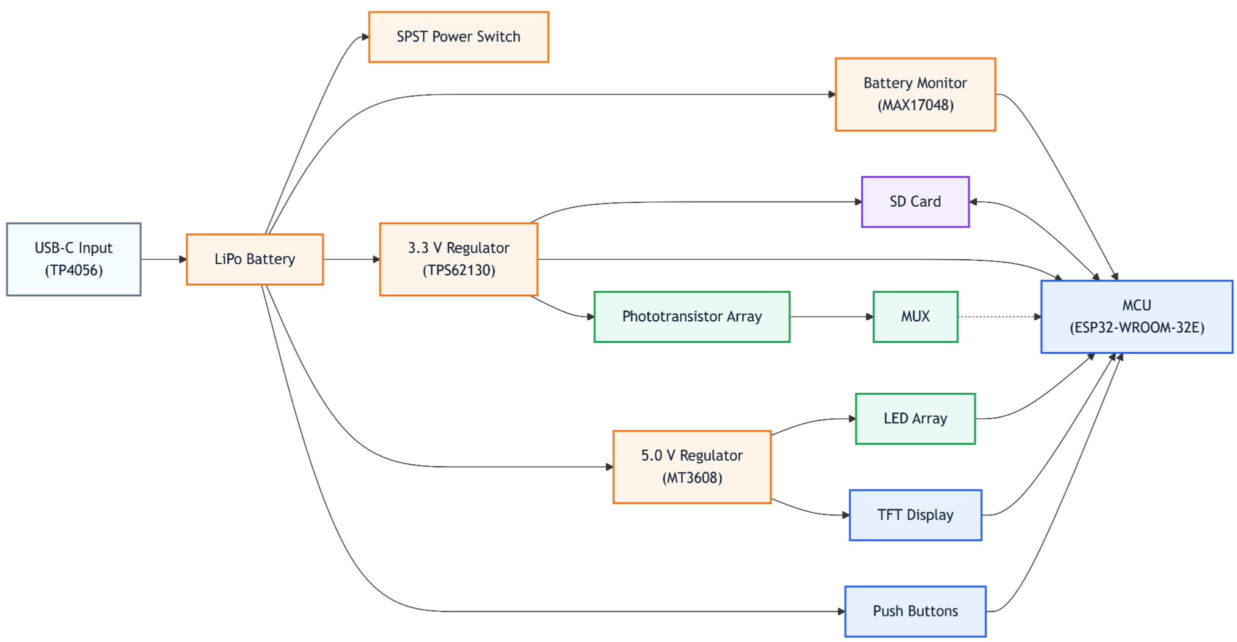


Figure 5.3: Preliminary Hardware Block Diagram for the Original LiPo-Based Design

5.1.3 Communication Interfaces

The system will use several communication interfaces to connect the major hardware blocks to the ESP32-S3 control module. The phototransistors will produce analog voltage signals in response to detected light. These signals will be routed through the multiplexer network, which will use digital GPIO lines for channel selection and enable control. The selected sensor output will then be read through one of the ESP32-S3's ADC inputs so the firmware can determine which sensor or group of sensors detected the laser strike.

The ST7796S TFT display will communicate with the ESP32-S3 through an SPI bus because this interface provides the speed required for graphical screen updates while reducing the number of connections compared to a parallel display interface. The FT6336U capacitive touch controller will use I2C communication and may share the same I2C bus with other lower-speed devices. Separate GPIO signals may also be used for display chip select, data/command, reset, backlight control, touch interrupt, and touch reset functions.

The physical buttons will connect directly to ESP32-S3 GPIO inputs and will be used for functions such as mode selection, score reset, calibration, and menu navigation. The microSD card will also use SPI communication for storing session records and system logs, with a dedicated chip-select signal allowing it to share the bus with the TFT display. Battery voltage and current data will be provided by the INA219 power monitor through I2C, allowing the firmware to estimate battery usage and display power information to the user.

Wireless communication between the target and companion mobile application will be handled through the ESP32-S3's built-in Bluetooth Low Energy hardware. BLE will carry shot data, scores, battery status, system settings, and training-mode commands without requiring a separate wireless module. Native USB Serial/JTAG connections will be used during development for firmware programming, serial monitoring, and debugging. All communication buses, control signals, and device connections will be clearly labeled in the schematic and PCB layout to simplify assembly, testing, and troubleshooting.

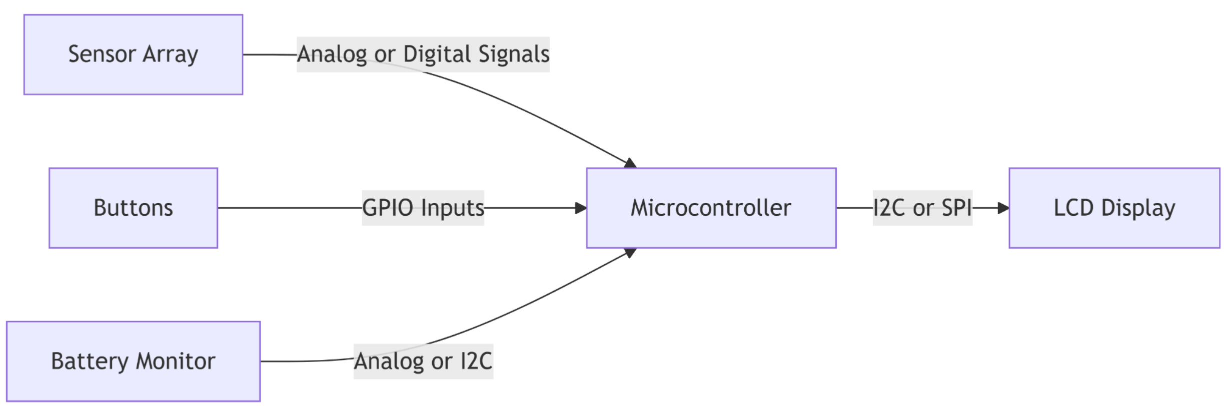


Figure 5.4: Hardware Communication Interface Diagram

5.2 Sensor and Feedback System

5.2.1 Sensor Array Design

The sensor array is one of the most critical subsystems in the PrecisionShot design, since it is the mechanism that determines whether a laser shot was detected and, if so, where on the target face it landed. Rather than treating each phototransistor as a simple binary trigger, the array is read as a set of analog voltage values, and the location of a shot is estimated by comparing the relative strength of the sensors nearest the impact point. A sensor located directly beneath the laser spot produces a larger photocurrent, and therefore a larger measured voltage, than sensors farther away. Therefore, the strongest sensor, combined with the weighted response of its immediate neighbors, is used to estimate the impact coordinate rather than reporting only which single sensor exceeded a threshold. The underlying operating principle that makes this possible—the proportional relationship between incident light and phototransistor output current—is detailed in Section 5.2.2.

The array uses 500 active phototransistor positions arranged in a uniform rectangular grid of 20 columns by 25 rows, matching the physical target layout described in Section 7.2. The sensors are evenly spaced across the active target area. The target's five concentric scoring zones are mapped over this grid, allowing the system to calculate shot position and score while maintaining consistent sensor spacing across the target face. This structure provides continuous sensing coverage without using the earlier 560-position mechanical-grid concept, reducing multiplexer, PCB, and per-unit sensor costs.

Because the array produces 500 individual analog signals, it cannot be wired directly to the ESP32-S3's limited ADC-capable pins. Section 5.3.1 describes the multiplexing architecture used to consolidate these 500 signals down to a small number of channels that the microcontroller can read, and Section 5.6.2 describes how the sensors are physically placed on the PCB relative to the front faceplate grid.

5.2.2 Phototransistor Operating Principles

Perhaps the most crucial component in our design is the phototransistor. The phototransistor is what allows the sensing of light to be translated into an electrical signal, but how exactly does that happen? Many specifications and design choices have been a direct result of the mechanisms of this component, thus it is necessary to understand it at the physical level.

A phototransistor combines a photosensitive semiconductor junction with transistor current gain. Incident light generates a small photocurrent in the base region, and the transistor amplifies that current into a larger collector current that can be measured by the sensing circuit [41]–[44]. Incidentally, this is exactly how photodiodes work, the major difference is what happens after the photocurrent is generated.

In a standard transistor, the base current is amplified into a significantly larger collector current by the current gain β (or hFE) which is usually around 100. In a phototransistor, this same amplification occurs, the difference being that the base is driven by photocurrent rather than being electrically connected directly. It is worth understanding that for our design this current gain is mostly irrelevant, and the datasheet for the component the team chose does not explicitly reveal this value. Instead, the datasheet provides a response curve that compares the illuminance in lux to the light current output. This is actually more useful than a nominal β value, since current gain is dependent on factors such as bias and temperature.

Phototransistors suffer from a small amount of leakage current, even when there is zero light, known as dark current. While this is typically only a few nanoamps, it is still amplified by the same current gain as any other transistor. Phototransistors also operate in two regimes, an active region where output current is proportional to light intensity, and a linear "saturated" region where the output clamps past a certain light level. Both of these flaws, while minor, are something that must be considered when designing shot location calculation methods.

It is also worth considering the use of a photodiode over the phototransistor, which has a number of benefits. Photodiodes lack the capacitive nature that an NPN transistor junction creates, this introduces a considerable delay that bare photodiodes do not suffer from. Photodiodes are also linear in their response, incident light tracks with output almost 1:1. A phototransistor relies on the collector current to create amplification, the gain of which is not fixed. Noise also can be a problem for the phototransistor as it will amplify any input signal, including dark current and ambient light. The obvious benefit that the phototransistor provided here is built in amplification, the photodiode alone would require a transimpedance amplifier which adds complexity to our design.

One more major characteristic of a phototransistor is its spectral response. The spectral response describes how sensitive the device is to different wavelengths of incident light. The energy of a photon is related to its wavelength according to the formula [94]:

$$E = hc / \lambda$$

In this formula, E is the energy of the photon, h is Planck's constant, c is the speed of light, and lambda (λ), is the wavelength. Different wavelengths, therefore, carry different amounts of energy. The semiconductor material and physical construction of the phototransistor, however, determine which wavelengths are most efficiently converted into electron-hole pairs and produce the highest photocurrent. Therefore, the sensitivity of a phototransistor isn’t constant across the entire visible spectrum [94[.

The spectral response of the selected phototransistor is very important for this application because the target system is designed to detect light from a specified laser. The phototransistor's datasheet provides a spectral response curve, which shows its sensitivity as a function of wavelength. The very peak of this curve represents the wavelength in which the device produces the greatest response for an amount of incident light. As the laser wavelength moves farther away from the peak response, the same optical power will generally produce a smaller electrical output. Because of this, the laser wavelength and phototransistor spectral response must be chosen together when designing the laser detection system [93], [94].

In addition to the wavelength of the laser, the spectral response also affects the system's susceptibility to ambient light. Ambient light, including sunlight and artificial indoor lighting contain light of different wavelengths, and intensity, and the phototransistor may respond to portions of this ambient illumination. Due to this ambient light, the sensor output contains a baseline signal, or voltage output, when the laser is not directly illuminating the sensor. The final signal-processing method must therefore distinguish being hit with the laser, and the baseline ambient light, or even slight changes in ambient light. The spectral-response characteristics of the selected phototransistor help to determine both the expected laser signal and the potential amount of ambient light to the sensor output [93].

At the physical level, the phototransistor operation begins with the interaction between incident photons and the chosen semiconductor material. When a photon with enough energy enters the semiconductor material, it can transfer its energy over to an electron in the valence band, allowing the electron to move into the conduction band. This process creates what is referred to as an electron-hole pair. In a phototransistor, the resulting carriers produce a current within the base region of the transistor. This current acts as the base current of a standard bipolar junction transistor, allowing the transistor action to amplify the resulting collector current [93].

The magnitude of the generated current depends mainly on the amount of light reaching the photosensitive junction. As the intensity of the incident light increases, more photons interact with the semiconductor material, and in turn generate additional electron-hole pairs. The resulting increase in current causes the collector current to increase. Therefore, when the phototransistor is operated in its active region, the electrical output is related to the intensity of the incident light. This behavior is very important for the design of the sensor array, because the system is not simply required to determine whether a phototransistor is illuminated or not [93]. Instead, the analog output voltage can be measured and compared between sensors to determine which portion of the array received the greatest amount of laser light. In other words, the amount of voltage across a resistor in series with the phototransistor will be measured, and the value of the voltage will allow the microcontroller to determine whether or not a laser hit that phototransistor. If only on or off states were being measured, each and every phototransistor would be in an on state, or 1 in binary.

The electrical behavior of the phototransistor is the same basic principle as the collector-emitter characteristic curves of a bipolar junction transistor. That being, at low levels of illumination, the photogenerated base current is small and the device operates in a region where the collector current increases with the amount of incident light. As the illumination increases, the collector current eventually becomes limited by the circuit it is a part of, and the available collector-emitter voltage. When this happens, the device approaches saturation and further increases in illumination produce a progressively smaller change in the output voltage. This is a very important consideration for the sensor array, because the signal-processing system must be able to distinguish between useful changes in illumination and the nonlinear behavior that occurs when a sensor is driven past its normal operating range.

For the PrecisionShot target system, the phototransistor is, therefore, used as an analog light sensor rather than as a simple digital switch. Each sensor is connected to a resistor that converts the phototransistor's current into a measurable voltage. The voltage from each sensor is then selected through the analog multiplexing system, and measured by the ESP32-S3's analog to digital converter. As stated earlier, by using this approach, it allows the system to preserve the relative intensity of the detected laser light from each sensor rather than reducing the output to a simple illuminated or non-illuminated state. The resulting analog data can then be used to determine the location of the laser impact across the sensor array.

5.2.3 Phototransistor Testing

To reliably detect the red laser signal used by PrecisionShot, the sensing subsystem required a visible-light phototransistor with strong sensitivity near the selected laser wavelength. The Everlight ALS-PT204-6C/L177 was selected for early bench testing because its spectral response peaks at approximately 630 nm and it is available in a convenient 3 mm through-hole package [12], [13]. Its spectral response peaks at 630 nm and rejects infrared wavelengths, making it well suited to detecting the red laser used for training rather than incidental IR sources found in many indoor environments. The Vishay TEPT5700 and TEPT4400 were also evaluated as through-hole alternatives because both provide broad visible-light sensitivity, although their spectral peaks are closer to 570 nm than the selected laser wavelength [14], [15].

Table 5.5: Candidate Through-Hole Phototransistors for Red Laser Detection

Part	Size	Spectral Peak/Range	Reasoning
ALS-PT204-6C/L177	3mm	630nm/550-700nm	Cheap, Mimics human eye response, rejects IR, analog output
TEPT5700	5mm	570nm/400-900nm	Less affordable than ALS, human eye response, wider range
TEPT4400	3mm	570nm/400-900nm	Similar to TEPT5700 but smaller, less expensive

The largest design consideration was ambient light. Because these phototransistors are tuned to mimic the response of the human eye, they can be triggered by any sufficiently bright visible-light source, not only the training laser. This is expected, since visible-spectrum sensitivity is necessary to detect the laser itself, but it also means the system must actively reject other sources of ambient light rather than relying on wavelength selectivity alone.

Bench testing was carried out by biasing each phototransistor through a series resistor from a bench power supply and reading the resulting voltage drop with a multimeter/oscilloscope. This was done first under normal indoor lighting alone, and then with the laser aimed directly at the sensor from a distance of several feet. Even bright white light sources produced substantially less output voltage than a direct red laser strike, confirming that a threshold-based approach could reliably separate genuine laser strikes from ambient light without relying on wavelength filtering alone.

This finding shaped the system's ambient light calibration approach. Rather than relying on a single fixed threshold chosen during bench testing, the target performs a full baseline scan across all 500 phototransistor sensors whenever the user presses the onboard ambient calibration button (Section 7.4), recording each sensor's current ambient level and updating detection thresholds accordingly. This allows the target to maintain reliable laser detection across a range of lighting conditions without requiring a firmware update or manual threshold entry for every new environment.

The wavelength match between the selected phototransistor and the Mantis Pink Rhino training laser, which emits in the 630 to 670 nm range (Section 5.8.2), further supports this sensor choice. Because the laser's wavelength sits close to the phototransistor's peak response, the resulting photocurrent, and therefore the measured voltage, is stronger than it would be with a laser operating farther from the sensor's peak sensitivity. This wavelength alignment was one of the key factors that carried the ALS-PT204-6C/L177 forward from the initial bench-testing phase into the SMD phototransistor selected for the production PCB, discussed next in Section 5.2.4.

5.2.4 SMD Phototransistors

The phototransistors used for early bench testing and breadboard validation are through-hole components, which are well suited to prototyping but impractical for a 500-sensor production board, both because of the board area 500 individually leaded parts would consume and because hand-soldering that many through-hole leads is not realistic for a single production run. The final sensor array PCB (Section 5.6.2) instead uses a surface-mount equivalent of the same phototransistor line.

The ALS-PT19-315C/L177/TR8 was selected as the surface-mount sensor because its visible-light spectral response and approximately 630 nm peak sensitivity closely match the ALS-PT204-6C/L177 used during bench testing, while its compact package is better suited for the full sensor-array PCB [10]. Beyond spectral response, package size and volume availability were the primary selection criteria for the production part. The ALS-PT19-315C/L177/TR8 does not cleanly fall into the standard 0603 footprint, but is only slightly larger, a tradeoff that was accepted after review since the part still fits within the pad spacing used in the sensor array layout and the corresponding aperture geometry of the front faceplate grid (Section 7.3).

This part is reflected in the current sensor array layout and faceplate alignment (Sections 5.6.2 and 7.3), and it is the SMD phototransistor the team is proceeding with for PCB fabrication. Because the SMD and through-hole versions share the same spectral characteristics, the calibration and threshold behavior established during bench testing with the through-hole part is expected to transfer directly to the final board. Additional confirmation is planned as part of the Senior Design 2 sensor bring-up and accuracy and latency testing milestones during week 7 (Table 9.2).

5.2.5 Visual Feedback Feedback Target Array

To enhance user interaction and provide immediate feedback, the target board features a localized visual indicator system. Each of the 500 individual phototransistor sensing cells is paired with a dedicated surface-mount light-emitting diode (LED). When a laser strike exceeds the detection threshold of a specific phototransistor, its adjacent LED illuminates with minimal delay. This provides direct real-time visual confirmation of the hit across the target face without requiring the user to look away or check the primary TFT LCD panel.

When designing an indicator matrix containing 500 individual LEDs, component packaging and manufacturing feasibility become critical considerations. Two primary package architectures were evaluated for the visual feedback array: standard Through-Hole (THT) LEDs and Surface-Mount Device (SMD) LEDs. Table 5.6 details the key engineering and manufacturing parameters compared between these two form factors.

Table 5.6: LED Package Form Factor Comparison for 500-Unit Feedback Array

Selection Criteria	Through-Hole (THT) LED (e.g., 3mm Radial)	Surface-Mount (SMD) LED (e.g., 1206 Package)
Package Dimensions	3.0 mm (Dia.) x 5.3 mm (H)	3.2 mm x 1.6 mm x 1.1 mm
Mounting / PCB Interface	Through-hole leads (Requires 1000 plated holes)	Surface-mount pads (Solder stencil paste)
Assembly Method	Manual hand-soldering or selective wave	Automated SMT Pick-and-Place machine
Assembly Time (500 Units)	High (~4 to 6 hours manual labor)	Very Low (~5 to 10 minutes automated placement)
Component Profile / Height	Tall (>= 7.0 mm including lead standoffs)	Low-profile (1.1 mm flush profile)
Reworkability & Alignment	Prone to height/tilt alignment variations	Self-aligns during reflow; high mechanical tolerance
Tape & Reel Packaging	Bulk or ammo pack (Requires lead trimming)	Standard EIA-481 8 mm Tape & Reel (3000 pcs/reel)

While through-hole LEDs offer simple breadboard prototyping, scaling to a 500-LED matrix makes THT assembly highly impractical due to the high pad count, drilling constraints, and excessive manual assembly time. SMD LEDs were chosen exclusively for this project. Utilizing an SMD package allows the team to leverage the university's automated SMT pick-and-place machine and reflow oven, drastically reducing manufacturing time, ensuring precise physical alignment beneath the 3D-printed grid, and minimizing overall PCB trace routing layer height.

To ensure high visibility against the target face while maintaining electrical compatibility with the system's power budget, several surface-mount LEDs were evaluated. Key selection criteria included unit cost at scale 500 units minimum with room for error of 15% so an additional 75 units, forward voltage compatibility with standard current-limiting driver topologies, luminous intensity, package size, and series consistency across multiple emission colors.

Table 5.7: Candidate Surface-Mount LED Specifications

Part Number	**Manufacturer**	**Package / Color**	**Peak Wavelength (λP)**	**Forward Voltage (VF Typ.)**	**Luminous Intensity (IV Typ.)**	**Max Continuous Current**	**Unit Cost @ 500+**

						(IF)	Qty
IN-S126ATB	Inolux	1206 SMD / Blue	470 nm	3.2 V @ 20 mA	150 mcd	25 mA	~$0.04
APT3216SURCK	Kingbright	1206 SMD / Hyper Red	645 nm	1.95 V @ 20 mA	200 mcd	30 mA	~$0.09
LTST-C191KGKT	Lite-On	0805 SMD / Green	571 nm	2.0 V @ 20 mA	35 mcd	30 mA	~$0.08
HSMW-C191	Broadcom	0805 SMD / White	Chromaticity (X=0.31, Y=0.31)	3.1 V @ 20 mA	180 mcd	20 mA	~$0.18

The Inolux IN-S126ATB (1206 Blue SMD LED) was selected as the visual feedback indicator for the PrecisionShot array. The 1206 package (3.2 mm x 1.6 mm) provides an ideal physical footprint that matches the adjacent phototransistor pitch while maintaining sufficient surface area for effortless automated pick-and-place operation and manual rework if needed. Furthermore, the IN-S126AT series offers identical mechanical pad layouts and footprint profiles across multiple color variations (such as Green IN-S126ATG and Yellow IN-S126ATY). This standard footprint allows future revisions of the target array to incorporate multi-colored visual feedback or scoring zones without altering the PCB footprint geometry. Combined with its exceptional cost efficiency under $0.05 per unit in bulk and high luminous output (150 mcd), the IN-S126ATB satisfies all optical, physical, and budgetary requirements for driving 500 indicator nodes.

5.2.6 User Interface Button Selection

To provide reliable manual control and system configuration, the PrecisionShot target enclosure features four dedicated user input switches mounted through the front faceplate. These four buttons execute primary operational commands: Power On/Off (or Wake), System Reset, Mode Change (Training Mode Selection), and Ambient Light Calibration.

A seamless physical and electrical user interface design requires careful consideration of mechanical alignment, user feedback, and electrical stability. The primary goal of this selection process is to identify switches that offer strong mechanical endurance against repetitive actuation while accommodating the exact physical geometry of the front enclosure panel.

Mechanical Integration and Dimension Constraints

The physical enclosure faceplate is manufactured with a nominal thickness of 6.1 mm and contains four circular cutouts, each 6.0 mm in diameter. To ensure ergonomic actuation and prevent flush-mount obstruction during operation, the switch plunger or actuator cap must extend through the full 6.1 mm panel depth and protrude approximately 1.5 mm to 2.5 mm beyond the outer faceplate surface.

If the actuator sits flush with or below the faceplate surface, pressing the button requires precise finger alignment or tools, significantly hindering usability during fast-paced target drills.

Conversely, excessive protrusion beyond 3.0 mm increases the risk of accidental actuation or mechanical shear if an object strikes the front face.

Two mounting topologies were evaluated to meet these physical constraints:

1. PCB-Mounted Extended-Actuator Switches: Switches are directly soldered to the main system PCB positioned behind the faceplate. While this eliminates internal wiring harnesses, it requires precise vertical PCB standoff spacing so that extended switch stems pass clean through the 6.0 mm faceplate holes without mechanical binding. Any tolerance stack-up across the PCB standoffs, board thickness variations, or slight PCB warping can cause switch stems to rub against the internal walls of the 6.0 mm cutouts, leading to sticking buttons or incomplete switch closures.
2. Panel-Mounted Threaded Bushing Switches: Switches mount directly into the 6.0 mm faceplate cutouts using a threaded subminiature bushing and hex nut assembly. They attach securely to the faceplate independent of PCB height tolerances and connect to the main controller board through a simple wiring harness or ribbon connector. This mechanical isolation decouples human actuation forces from the internal printed circuit board assembly, ensuring long-term structural reliability.

Table 5.8: User Interface Control Button Mapping and Functional Description

Button Function	Input Type	Microcontroller Interface	Operational Behavior
Power Switch	Latching / Soft-Power Momentary	ESP32-S3 GPIO (Wake / Power Rail)	Toggles main system power state or enters deep sleep
System Reset	Momentary Pushbutton	ESP32-S3 EN / Hardware NRST Line	Triggers hard hardware reset of the core processor
Mode Change	Momentary Pushbutton	ESP32-S3 GPIO (Interrupt-driven)	Cycles between target scoring routines and drill modes
Ambient Light Calibration	Momentary Pushbutton	ESP32-S3 GPIO (Interrupt-driven)	Triggers baseline light sampling across all 500 sensors
Mode Change	Momentary Pushbutton	ESP32-S3 GPIO (Interrupt-driven)	Cycles between target scoring routines and drill modes

Ambient Light Calibration	Momentary Pushbutton	ESP32-S3 GPIO (Interrupt-driven)	Triggers baseline light sampling across all 500 sensors

Component Selection and Candidate Comparison

Four subminiature switch candidates available through DigiKey were evaluated against the physical dimensions 6.0 mm hole diameter, 6.1 mm panel thickness, electrical current requirements, tactile feedback, and structural mounting durability.

The evaluation focused on two primary metrics: mechanical endurance and mounting versatility. PCB-mounted options were analyzed for stem length and standoff clearance, while panel-mounted options were evaluated based on bushing size, plunger throw distance, and thread engagement depth across the 6.1 mm faceplate thickness.

Table 5.9: Candidate Control Button Specifications and Selection Matrix

Part Number	Manufacturer	Mounting Type	Bushing / Plunger Specs	Panel Cutout Fit (6.0 mm Target)	Unit Cost @ 10+ Qty
EB2011-BG	NKK Switches	Panel Mount Threaded	1/4" (6.35 mm) Bushing / 8.1 mm Plunger	Direct Panel Mounting (M6 / 1/4" Hole)	~$3.80
700SP7B11M6REH	E-Switch	Panel Mount Subminiature	6.35 mm Bushing / 7.9 mm Plunger	Direct Panel Mounting (M6 / 1/4" Hole)	~$4.00
B3F-1025	Omron	PCB Through-Hole	13.0 mm Total Stem Height	Requires 4.5 mm PCB Standoff Offset	~$0.45
PTS645VL132LFS	C&K	PCB Surface-Mount	13.0 mm Extended Stem	Requires 4.5 mm PCB Standoff Offset	~$0.38

The NKK Switches EB2011-BG was selected as the primary user interface switch. The EB2011 series features a 1/4-inch (6.35 mm) threaded bushing that secures directly through the 6.0 mm faceplate cutouts with slight reaming or standard 1/4" sizing using an included locking hex nut and washer.

With an 8.1 mm overall plunger and cap height, the button extends through the entire 6.1 mm thick faceplate to provide a comfortable 2.0 mm actuator protrusion beyond the front surface. Panel mounting isolates the mechanical pressing force from the main internal PCB, preventing mechanical stress, micro-fractures, and fatigue on solder joints over repetitive user operations.

While the unit cost of panel-mounted switches like the EB2011-BG is higher than basic tactile switches, the mechanical feedback, long rotational life, and elimination of delicate PCB standoff height dependencies justify the selection for a hardware user interface.

5.3 Multiplexing

5.3.1 Multiplexing Architecture

The phototransistor array must feed its 500 individual analog signals into the ESP32-S3 in order for the firmware to calculate shot placement, but the microcontroller can only interface with a limited number of GPIO and ADC-capable pins. Connecting each sensor directly to its own ADC input is not possible at this scale, so the system relies on multiplexing to reduce the number of signals that must reach the microcontroller directly.

An analog multiplexer functions conceptually as a digitally controlled rotary switch. A 16-channel multiplexer, for example, can select among 16 separate analog inputs using 4 digital select lines, since 4 binary select pins are enough to address 16 unique channels. Each phototransistor in the array is wired to one input of a multiplexer, and the multiplexer's digital select lines determine which single sensor's voltage is passed through to its output at any given moment. That output is then read by one of the ESP32-S3's ADC inputs, digitized, and passed to the firmware, which already knows which physical sensor corresponds to that select-line combination and can therefore map the reading back to a location on the target.

Two general architectures were considered for how these multiplexers connect to the ESP32-S3's ADC inputs. The first is a single-stage design, in which every multiplexer's output is wired directly to its own dedicated ADC pin. This is the more straightforward implementation, but because covering 500 sensors requires multiple 16-channel or 32-channel multiplexers, it also consumes a correspondingly large share of the microcontroller's limited ADC-capable pins. The second approach is a cascaded, or tiered, design, in which the outputs of a first stage of multiplexers are fed into a second stage rather than into the ADC directly. This reduces the number of ADC pins required, since many first-stage multiplexers can ultimately be consolidated onto just one or two ADC channels, but it introduces additional series resistance and settling delay at each added stage, which can degrade signal integrity if not accounted for.

The tradeoff between these two approaches, along with the specific multiplexer parts evaluated for the design, is discussed in Section 5.3.2. The resulting scan-timing analysis, including how much delay the cascaded approach actually introduces relative to the 10 ms LED feedback objective, is covered in Section 5.3.3.

5.3.2 MUX Selection

Section 5.3.1 introduced the 16-channel multiplexer as a representative example, but several other multiplexer parts were evaluated before a final selection was made. This section compares three candidate devices and the criteria that mattered most for the PrecisionShot sensor array: channel count, on-resistance, switching speed, supply compatibility, cost, and library or documentation support.

ADG732 - a monolithic CMOS 32-channel analog multiplexer that uses a 5-bit binary address to control switching. Because each device covers twice as many channels as a 16-channel part, fewer multiplexers are needed overall to cover the full 500-sensor array. The ADG732 also offers superior electrical specifications compared with the 16-channel options considered, including a lower on-resistance that helps preserve signal integrity as the analog voltage passes through the switch. Its primary drawback is cost. Each 32-channel device is significantly more expensive than a 16-channel alternative, even though it only does the work of two of them.

ADG1606 - a monolithic iCMOS 16-channel analog multiplexer that uses the same 4-bit binary addressing scheme described in Section 5.3.1. Its main advantage is an on-resistance that stays low and remains flat across the input voltage range the design operates in, meaning the resistance seen by a given signal does not shift meaningfully with the voltage being switched. Of the three parts considered, the ADG1606 provides the strongest signal-preservation performance, which becomes important if resistance variation in a cheaper multiplexer turns out to distort the sensor readings. This performance comes at a higher per-unit cost than the other two candidates.

CD74HC4067 – The Texas Instruments CD74HC4067 is a 16-channel CMOS analog multiplexer that is widely documented and supported by existing libraries and firmware examples [8], [9]. It uses the same 4-bit addressing scheme as the ADG1606, making the two parts architecturally interchangeable if a substitution becomes necessary later in development. The 74HC4067 has a considerably faster channel-switching time than the ADG1606, which leaves more of the 10 ms feedback budget available for other stages of the signal path, and it is also the least expensive of the three parts. Its main drawback is a non-linear on-resistance that varies across different input voltages rather than remaining flat.

Table 5.10: Analog Multiplexer Comparison for Sensor Array Scanning

Criteria	74HC4067	ADG1606	ADG732
# Channels	16	16	32
On-resistance (RON)	70Ω typ @ 4.5V, likely higher for 3.3V	14Ω typ @ 3.3V	~7-11Ω typ @ 3V
RON flatness	Not flat, varies	5Ω @ 3.3V	~1Ω typ @ 3V
Channel switching time	60ns typ @ 4.5V	353ns typ @ 3.3V	34ns typ @ 3V
Minimum muxes required, 500 sensors	32	32	16
Supply Voltage Range	2-6V	3.3-16V single; ±3.3–±8V dual	1.8-5.5 V single; ±2.5 V dual
Relative Cost	Low (~$1 ea.)	High (~$5 ea.)	High (~$7 ea.)
Library Support	Extensive	Limited	Limited
Best Use Case	Speed, cost	Analog fidelity	Channel density

Based on this comparison, the 74HC4067 was selected as the multiplexer for the PrecisionShot sensor array. The deciding factor was cost. At the 500-sensor scale required by this design, the price difference between the 74HC4067 and either alternative would meaningfully exceed the project's component budget (Section 8.2) without a clear performance benefit for this application. This does not rule out the ADG1606 entirely. If bench testing during Senior Design 2 shows that the 74HC4067's non-linear on-resistance introduces enough error to distort sensor readings, the ADG1606 remains a compatible drop-in alternative given its matching 4-bit addressing scheme.

5.3.3 MUX Implementation

There are a few different options for how we implement the multiplexers in our design. We need to consider the constraints introduced by the ESP32 we chose as well as the 10 ms LED feedback objective. The ESP32-S3-WROOM-1-N8R8 has 2 SAR ADC cores, each with a singular SAR converter and 10 channels dedicated to each, giving us a total of 20 ADC capable channels. It is also worth noting that ADC2 shares its circuitry with the Wi-Fi radio, however our current design relies exclusively on BLE for wireless connections.

Our first option is using a single stage mux design, with each individual mux output wired directly to its own dedicated ADC channel on the ESP32. As discussed in section 5.3.2, a 16 channel multiplexer will require at minimum 32 muxes to cover the 500 phototransistors. Since the ESP32-S3-WROOM-1-N8R8 only has 20 channels capable of ADC, this immediately removes the 74HC4067 and ADG1606 as candidates for this design. This leaves us with the ADG732 32 channel mux as our only option for this implementation. However, with 32 channels we would still need 16 ADC pins of the MCUs 20, leaving us little headspace for other ADC requiring components.

Our likely option is the cascaded architecture, which would route the initial multiplexing stage into a secondary MUX tier. For example, the 16 channel MUX with 32 output channels could be consolidated into just 2 channels with the addition of 2 multiplexers after the first stage. This would preserve many ADC pins and would allow each of the 2 channels to have its own ADC core. The downside of this design is that adding additional components creates resistance and capacitance resulting in propagation delay. We can model the settling time of two cascaded 74HC4067 with an RC estimate. The 74HC4067 has an R_{ON} of 70 Ω, and a C_{COM} of 50pF, $\tau = RC = 70\ \Omega \times 55\ \text{pF} \approx 3.9$ ns. For a 12-bit ADC, we can use the following equation to estimate the number of time constants needed for the exponential settling error to drop to ½ LSB: $N_{TC} = (N+1)\cdot\ln(2)$. Plugging N=12 gives us $N_{TC} = (12 + 1) \times \ln(2) = 13 \times 0.6931 \approx 9.01$, this gives us a delay of 9×3.9 ns ≈ 35 ns per stage. SAR ADC acquisition time must allow the switched input signal to settle sufficiently close to its final value before conversion. Acquisition-time analysis commonly models this behavior using the source resistance, input capacitance, ADC resolution, and the number of RC time constants required to reach the desired error limit [47], [48]. For the preliminary 12-bit estimate used here, approximately nine time constants are required to reduce the remaining settling error below one-half of an LSB. This delay happens once for each stage giving us a total delay of about 70 ns before entering the ESP32. While this delay is comparable to the 60 ns switching time, this is still significantly faster than the time it takes for a single ADC conversion, which is on the order of microseconds.

The internal-ADC architecture of the ESP32-S3's SAR core is what ultimately limits the speed at which we can produce readings, this can vary significantly depending on the method of communication with the firmware. Espressif's ESP32-S3 ADC documentation describes both one-shot and continuous sampling modes, with the continuous driver supporting configured sampling rates up to approximately 83.3 kS/s under the documented operating limits [49]. However, the simple one-shot API that most firmware uses carries extra overhead from mutex locking, attenuation handling, and register setup. Espressif provides the ESP-IDF ADC driver APIs used to configure one-shot and continuous analog sampling on the ESP32-S3 [5]. The Arduino and ESP-IDF environments provide convenient ADC read functions, but their software overhead can produce substantially slower sampling than the ADC hardware limit. Documentation and developer measurements indicate that standard one-shot and analogRead() paths may require tens or hundreds of microseconds per reading depending on configuration and implementation [50], [51], [53]. A direct-register ESP32-S3 ADC implementation has demonstrated sampling near 110 kS/s, or approximately 9 µs per reading, by avoiding part of the overhead associated with the standard high-level ADC functions [52].

Table 5.11: ESP32-S3 Internal ADC Throughput Scenarios

Scenario	Approx. Time / Sample	Approx. Rate
Stock one-shot driver (adc_oneshot_read / analogRead)	≈100 µs	≈10,000–16,000 sps
Continuous-mode hardware ceiling	≈12 µs	83,333 sps
Hand-optimized direct register access	≈9–10 µs	≈110,000 sps

This leads us into the external ADC options, the first of which is utilizing parallel ADC banks. A parallel-bank architecture replaces the internal ADC entirely with several external ADC chips that each contain their own analog multiplexer, split across independent SPI buses so multiple chips can be read at the same time. In addition to the interfaces reserved for internal flash and PSRAM, the ESP32-S3 provides two general-purpose SPI controller options that may be used to divide external ADC, display, or storage traffic across separate buses [55]. Both remain fully available even with the Octal PSRAM used on the N8R8 module, which only reserves GPIO33–GPIO37 for the PSRAM link and does not consume either general-purpose SPI peripheral.

A representative implementation uses 32 Texas Instruments ADS7953 converters 12-bit, 16-channel, single-ended SAR ADCs with a built-in auto-sequencing multiplexer and a specified 1 MSPS throughput, split evenly with 16 chips on SPI2 and 16 on SPI3, with each chip's CS line driven by a spare GPIO (native hardware CS count is a convenience, not a hard limit, since CS can also be bit-banged). Because the ADS7953 already contains its own mux, no separate 74-series or ADG-series multiplexer chip is needed for this architecture at all. One possible parallel architecture would use Texas Instruments ADS7953 devices, which are 12-bit, 16-channel, single-ended SAR ADCs with an internal input multiplexer, serial interface, automatic channel sequencing, and maximum throughput of 1 MSPS [56], [57].

The other promising external ADC option uses just a single ADC on one SPI bus, rather than splitting multiple chips across two busses. This is much simpler compared to the other external design at the cost of losing bus parallelism, meaning this design is more dependant on the hardware device. The MAX11128 is another candidate because it combines a 12-bit SAR ADC, 16-channel input multiplexer, scan-mode sequencer, maximum 1 MSPS conversion rate, and a 2.35 V to 3.6 V supply range compatible with the system's 3.3 V rail [58], [59]. The MCP3208 provides eight single-ended input channels, 12-bit resolution, and an SPI interface, with its maximum sampling rate dependent on supply voltage and SPI clock configuration [60], [61]. At the project's 3.3 V rail, its achievable conversion rate is expected to be lower than the 100 kS/s rating specified under 5 V operating conditions and must be confirmed through bench testing.

Table 5.12: ADC Device Comparison for the Scanning Architecture Study

Part	Type	Channels / chip	Max Rate	Interface	Internal Mux?	Supply	Approx. Cost Ea.
MCP3208	SAR ADC	8	~63 ksps @3.3V	SPI	Yes (8:1)	2.7–5.5 V	~$4–$5
ADS7953	SAR ADC	16	1 MSPS	SPI, auto-sequencer	Yes (16:1)	2.7–5.25 V	~$9–$11
MAX11128	SAR ADC	16	1 MSPS	SPI, scan mode	Yes (16:1)	2.35–3.6 V	~$8–$10 (est.)

Combining the mux settling times from Table 5.10, the ADC throughput scenarios from Table 5.11, and the device throughput from Table 5.12, we estimated the time to read all 500 sensors once under each architecture. These are engineering estimates built from datasheet parameters and published benchmarks, not bench measurements; validating them against real hardware is exactly the purpose of the SD2 "Accuracy and latency testing" milestone in Table 9.2, and the figures below should be treated as planning numbers rather than guaranteed results.

Table 5.13: Estimated 500-Channel Scan Time by Architecture

Architecture	ADC Approach	Chips / Buses Used	Per-Channel Budget	Total Scan Time	Effective Refresh Rate	Meets 10 ms objective?
1: Single-stage MUX	Internal ADC, stock driver, ADC1 only	16× ADG732, 16 ADC1/2 pins	~100 µs	~50 ms	~20 Hz	No
1: Single-stage MUX	Internal ADC, stock driver, ADC1+ADC2 split	16× ADG732, 16 ADC1/2 pins	~100 µs	~25 ms	~40 Hz	No
1: Single-stage MUX	Internal ADC, optimized driver, ADC1+ADC2 split	16× ADG732, 16 ADC1/2 pins	~12 µs	~3.0 ms	~333 Hz	Yes

2: Cascaded MUX	Internal ADC, optimized driver, ADC1+ADC2 split	32+2 mux, only 2 ADC pins	~12 µs	~3.0 ms	~333 Hz	Yes
3: Parallel ADC banks	32× ADS7953 across SPI2 + SPI3	32 chips, 2 SPI buses, 0 ADC pins	~1 µs	~0.26 ms	~3,900 Hz	Yes, large margin
4: External ADC (single, mux-fed)	1× MAX11128 + 16× ADG732, 1 SPI bus	17 chips, 1 SPI bus, 0 ADC pins	~1.03 µs	~0.52 ms	~1,940 Hz	Yes
4: External ADC (single, native mux)	63× MCP3208, 1 SPI bus, sequential	63 chips, 1 SPI bus, 0 ADC pins	~15.9 µs	~7.95 ms	~126 Hz	Yes, tighter margin

Two results fall out of this comparison. First, the choice between single-stage and cascaded multiplexing barely affects scan speed at all. Both land at roughly the same total in Table 5.13, because the analog switching time of any of the three multiplexers is dwarfed by the ADC's own conversion time. The real reason to prefer the cascaded design from Section 5.3.1 is the ADC pin budget: it reaches all 500 sensors on 2 ADC-capable pins instead of 16, leaving the rest of ADC1 and ADC2 free for the ambient light sensor and other analog signals, at a settling-time cost measured in tens of nanoseconds.

Second, the internal ADC's own driver overhead, not the multiplexer topology, is what puts the 10 ms LED response objective at risk. A stock-driver, single-core implementation misses that budget by roughly 5×; splitting the read across ADC1 and ADC2, or bypassing the standard driver for direct register access, is enough to clear it comfortably. Migrating to an external ADC removes this dependency altogether: a single MAX11128 behind the existing MUX front end, or a bank of ADS7953 devices split across the ESP32-S3's two free general-purpose SPI buses, scans all 500 channels in under half a millisecond, roughly an order of magnitude faster than even an optimized internal-ADC read, while freeing every internal ADC pin for other analog subsystems. Given the project's budget constraints from Chapter 8, we are still considering the cascaded internal-ADC design described in Section 5.3.1 as the baseline option. We can pair this with a hand-optimized ADC read path to guarantee the 10 ms feedback target. In a case where the internal ADC's noise or timing margin is not meeting our standards during prototyping, we can transition to an external ADC design utilizing either the MAX11128 or ADS7953.

5.4 Processing and Control System

5.4.1 MCU Module

The microcontroller selected for the system must support the required communication interfaces and be able to connect with each hardware device. Since the companion mobile application will communicate directly with the MCU, Bluetooth Low Energy capability is required. Wi-Fi support is available for possible future expansion but is not required for normal system operation. Another important consideration is power draw. The MCU will remain operational whenever the device is powered, so current consumption must be minimized to avoid unnecessarily reducing battery life. Based on these requirements, the

ESP32-S3-WROOM-1-N8R8 was selected because it satisfies the system's communication, processing, memory, and power requirements.

The ESP32-S3-WROOM-1-N8R8 combines processing, memory, wireless communication, and embedded peripherals in one compact module. The ESP32-S3 family uses a dual-core 32-bit Xtensa LX7 processor operating at up to 240 MHz and provides the GPIO, ADC, SPI, I2C, USB, timer, and wireless functions required by the project [45], [46]. This provides enough processing capability to handle sensor scanning, shot detection, scoring logic, display updates, Bluetooth Low Energy communication, and debugging without requiring a separate processor. The selected N8R8 variant also includes 8 MB of flash memory and 8 MB of PSRAM, giving the firmware additional space for program storage, sensor buffers, display graphics, BLE packet handling, and future software expansion.

The module also supports a large number of useful embedded peripherals for the PrecisionShot system. These include GPIO, ADC, SPI, I2C, UART, LCD interface support, LED PWM, USB Serial/JTAG, SD/MMC host support, timers, and watchdogs. This is important because the MCU must communicate with several different subsystems, including the sensor multiplexer network, TFT display, capacitive touch controller, physical buttons, microSD card storage, LED feedback hardware, battery-monitoring hardware, and the companion mobile application. The ESP32-S3-WROOM-1-N8R8 also operates from a 3.0 V to 3.6 V supply, which matches the planned 3.3 V system rail and makes it compatible with the rest of the low-voltage digital hardware.

5.4.2 Microcontroller Selection

Our original MCU was planned to be the ESP32-WROOM-32E. This has since shifted to the ESP32-S3-WROOM-1-N8R8, which more neatly fits our requirements. The most notable advantage is the Bluetooth capability. The N8R8 supports BLE, or Bluetooth Low Energy, instead of relying on Bluetooth Classic. Bluetooth Classic was never an important aspect of the MCU because our design does not require continuous audio-style signal output. Our use case will benefit more from the high bandwidth options and lower power operation available through BLE.

The ESP32-S3-WROOM-1-N8R8 also provides a stronger development path than the original ESP32-WROOM-32E option. Since PrecisionShot requires real-time sensor processing and app communication, the selected module gives the team more room to grow as the firmware becomes more complex. The extra PSRAM is especially useful for buffering sensor readings, handling display elements, storing temporary shot data, and supporting future debug tools. This makes the S3 module a better fit for a system that is expected to handle embedded control, wireless communication, and local user feedback at the same time.

Another major reason for selecting the ESP32-S3-WROOM-1-N8R8 is its built-in USB Serial/JTAG controller. This allows the team to program and debug the module more directly during development, reducing the need for extra USB-to-UART bridge circuitry on the final PCB. This simplifies the control board design, lowers component count, and makes firmware testing easier during prototyping. Since Senior Design II will require repeated firmware updates, debugging, and hardware testing, this built-in development support reduces design risk and makes the module more practical for the team's timeline.

5.4.3 Programming and Debug

As mentioned previously, the onboard USB Serial/JTAG controller allows the team to program and debug the microcontroller without requiring a separate USB-to-UART bridge circuit. Other important considerations include the boot, enable, and strapping connections that determine how the ESP32-S3 starts and whether it enters normal operation or firmware-download mode.

The ESP32-S3-WROOM-1-N8R8 includes a native USB Serial/JTAG controller, which simplifies firmware development by allowing the team to upload code, monitor serial output, and use debugging tools directly through a USB-C connection. The USB D+ and D− signals will be routed between the ESP32-S3 and the programming connector, while the required USB-C configuration resistors will be included so the connection is recognized correctly by a host computer. Accessible test points should also be provided for USB D+, USB D−, 3.3 V, and ground to simplify troubleshooting during PCB bring-up.

The firmware will require frequent updates as the sensor-array scanning, multiplexer control, shot-detection logic, scoring, calibration, LED feedback, TFT display, microSD logging, and Bluetooth Low Energy communication are developed. Serial logging will be used to display sensor readings, selected multiplexer channels, detected shot coordinates, scoring results, active training modes, battery data, storage status, and BLE connection events. JTAG debugging will allow the team to pause execution, inspect variables, set breakpoints, and identify timing or logic problems that may not be visible through serial messages alone.

Programming and debugging will also support integration with the companion mobile application. The ESP32-S3 firmware will provide BLE services for transmitting shot data, score information, battery status, active mode, and system-state updates to the app. It will also receive commands from the app for mode selection, distance settings, calibration, session reset, and other supported controls. During development, serial logs and BLE packet monitoring can be used together to verify that data generated by the firmware matches the information displayed and stored by the React Native application. This will help isolate whether an error originates in the sensor-processing firmware, BLE communication layer, packet format, or mobile-app logic.

The PCB will include an enable/reset button connected to the ESP32-S3 EN pin and a boot button connected to GPIO0. These controls will allow the device to restart, enter firmware-download mode, and recover if a firmware error prevents normal startup. The required pull-up and pull-down resistors for the enable, boot, and other strapping pins must be selected and routed according to the ESP32-S3 hardware requirements so the module consistently enters the intended operating mode. Test points for EN, GPIO0, and other important boot signals will also make it easier to diagnose startup and programming failures.

Together, the native USB connection, boot and reset controls, serial monitoring, JTAG debugging, and BLE integration tools will provide a complete development path for both firmware testing and mobile-app communication. These features are especially important during prototype development because the embedded firmware and companion application must be tested repeatedly as individual subsystems are added and integrated.

5.4.4 GPIO Allocation

Table 5.14: Preliminary ESP32-S3 GPIO Allocation by Subsystem

Subsystem	Interface Type	Required Signals	Estimated Pin Count	Current Prototype Assignment
MUX select lines	Digital GPIO	S0, S1, S2, S3	4	TBD
MUX enable / bank selection	Digital GPIO	Enable and bank-select signals	TBD	TBD
MUX outputs	ADC	Analog sensor outputs	TBD	TBD
ST7796S TFT display	SPI + digital GPIO	SCLK, MOSI, MISO, CS, DC, reset, backlight	7	GPIO11 SCLK, GPIO12 MOSI, GPIO9 MISO, GPIO3 CS, GPIO13 DC, GPIO14 reset, GPIO10 backlight
FT6336U touch controller	I2C + digital GPIO	SDA, SCL, interrupt, reset	4	GPIO6 SDA, GPIO8 SCL, GPIO5 interrupt, GPIO7 reset
microSD card	Shared SPI	SCLK, MOSI, MISO, dedicated CS	1 additional pin when sharing SPI	GPIO4 CS; shares GPIO9, GPIO11, and GPIO12
Battery monitor	Shared I2C	SDA, SCL, optional alert	0–1 additional pins	Shares GPIO6 SDA and GPIO8 SCL; alert TBD
Ambient-light sensor	Shared I2C	SDA, SCL, optional interrupt	0–1 additional pins	Optional; shares the I2C bus
Physical push buttons	Digital GPIO	Mode, reset, calibration, and navigation inputs	3–6	TBD; current prototype primarily uses touchscreen controls
Regulator Power Good	Digital GPIO input	LMR33620 PG status	1	TBD
Boot / enable	Boot and reset control	GPIO0 boot and EN/reset	2	Final schematic assignment required
Native USB programming	USB Serial/JTAG	USB D+ and USB D−	2 dedicated USB signals	Final schematic assignment required
Bluetooth Low Energy	Internal ESP32-S3 radio	BLE services and characteristics	0	No external GPIO required

The GPIO allocation shown in Table 5.14 remains preliminary and will be finalized using the ESP32-S3 GPIO capabilities, reserved functions, strapping-pin requirements, and peripheral-routing restrictions documented by Espressif [54]. The current prototype firmware already defines the principal connections for the TFT display, capacitive touch controller, and microSD card. The ST7796S display uses the SPI bus with GPIO11 for SCLK, GPIO12 for MOSI, and GPIO9 for MISO. Separate control signals are assigned to GPIO3 for display chip select, GPIO13 for data/command selection, GPIO14 for display reset, and GPIO10 for backlight control. The microSD card shares the same SPI clock and data lines while using GPIO4 as its dedicated chip-select signal.

The FT6336U capacitive touch controller uses a separate I2C connection, with GPIO6 assigned to SDA and GPIO8 assigned to SCL. GPIO5 is used for the touch-interrupt signal, and GPIO7 controls the touch-controller reset line. The current firmware initializes this I2C bus at 400 kHz and uses the touch controller as the primary local input method for menu navigation, session reset, BLE controls, and display settings.

The ESP32-S3-WROOM-1-N8R8 provides GPIO, ADC, SPI, I2C, UART, USB Serial/JTAG, and PWM functionality, making it capable of supporting the major PrecisionShot subsystems. However, careful pin planning is still required because the sensor multiplexer network, display, touch controller, storage, physical buttons, battery monitoring, power-status signals, and debugging connections must share the available resources. The highest-priority assignments will be reserved for the sensor multiplexing architecture because its select lines, enable lines, and ADC inputs directly affect shot-detection speed and reliability.

The TFT display and microSD card can share the SPI clock and data signals because each device uses a separate chip-select line. Similarly, the touch controller, INA219 battery monitor, and optional ambient-light sensor can share the I2C bus as long as each device has a unique address. Additional GPIO signals may still be required for interrupts, reset control, regulator Power Good monitoring, and physical buttons.

Bluetooth Low Energy communication does not require external GPIO assignments because the radio is integrated into the ESP32-S3. The firmware currently implements a BLE GATT server that transmits test-hit and score data to the companion application while receiving commands such as session reset and connection checks. Native USB Serial/JTAG signals will be reserved for firmware programming, serial monitoring, and debugging. Exact boot, enable, USB, multiplexer, button, and remaining peripheral assignments will be finalized in the ESP32-S3 schematic before PCB routing begins.

5.5 Display and Physical User Interface

5.5.1 Display Requirements

An important part of the PrecisionShot user interface is the onboard display. The display must provide consistent real-time feedback so the user can see system status, active mode, shot location, score, calibration information, and other relevant training data without relying entirely on the mobile application. Since the target is designed to operate as a standalone device, the display must be clear enough to support normal use even when the app is not connected.

For this application, the graphical output must be detailed enough to show a rendered target area with visible shot placements from a reasonable distance. The display should be readable indoors and outdoors, while also keeping power consumption low enough to support battery-powered operation. It must also interface cleanly with the selected ESP32-S3-WROOM-1-N8R8 microcontroller without requiring unnecessary support hardware. Cost, size, color output, refresh rate, and ease of software integration are also important factors because the display must fit within the enclosure and remain practical for the project budget.

Several display technologies were considered for the onboard user interface. Each option has advantages and disadvantages related to cost, power draw, readability, graphical capability, and ESP32 compatibility. The following section compares the most relevant display types and explains why a TFT LCD is the best fit for this system.

5.5.2 Display Technology Comparison

Character LCD - A character LCD is a monochrome display that uses fixed text cells. It is simple, inexpensive, and easy to program, but it does not meet the graphical needs of the PrecisionShot system. Since it can only display letters, numbers, and symbols, it cannot show a rendered target or precise shot placement. This makes it unsuitable for the onboard visual feedback required by this project.

Passive Graphic LCD - A passive graphic LCD uses a monochrome electrode matrix to display basic graphics. This type of display is attractive because it is low cost and has very low power consumption. However, passive graphic LCDs usually have lower refresh rates, limited contrast, and lower resolutions. These limitations make them a poor fit for displaying active shot feedback, especially when the system may need to show many possible shot locations across the target area.

Vacuum Fluorescent Display - A vacuum fluorescent display is a bright monochrome display that uses cathodoluminescence. It offers strong visibility, including in bright environments, but it is not ideal for this project. VFDs are typically better suited for text, numbers, and simple indicators rather than detailed graphical target rendering. They also have higher power consumption than other display options, which makes them less practical for a battery-powered training device.

OLED - An OLED display uses self-illuminating pixels and provides excellent image quality, high contrast, true blacks, and strong color reproduction. These characteristics would create a high-quality user experience, but larger color OLED displays are expensive compared to other options. Smaller or monochrome OLED modules are more affordable, but they do not provide the size or graphical capability needed for this system. OLED displays may also be more vulnerable to long-term wear or burn-in, especially if static interface elements remain on screen for extended periods.

TFT LCD - A TFT LCD is an active matrix display that uses a transistor and capacitor at each pixel. This technology supports full-color graphics, comes in a wide range of sizes, and is much more affordable than larger OLED displays. TFT LCDs also avoid OLED burn-in concerns and are commonly used in embedded systems. The main drawbacks are backlight power consumption, lower contrast than OLED, and narrower viewing angles depending on the panel type. Even with these limitations, TFT LCD technology provides the best overall balance of cost, graphical capability, size, availability, and ESP32 compatibility for the PrecisionShot system.

Table 5.15: Display Technology Comparison for the Onboard User Interface

Criterion	Character LCD	Passive Graphic LCD	VFD	OLED	TFT LCD
Shot plot	No, Text Only	Yes, low quality	Yes, low quality	Yes	Yes
Color	Monochrome	Usually Monochrome	Monochrome	Yes, costs more	Yes
Size	Small	Varies	Small	Varies, larger is expensive	Varies, affordable
Cost	Cheapest	Cheap	Varies	Expensive	Affordable

Daylight readability	Good	Good	Excellent	Poor	Fair
Power	Very low	Very low	High	Low	Low
Refresh rate	Slow	Slow	Moderate	Fast	Fast
Burn-in risk	No	No	Some	Yes	No
ESP32 Support	Yes	Yes, depends	Yes, but poor	Yes	Yes
Viewing angle	Fair	Poor	Wide	Wide	Fair

Overall, TFT LCD technology is the most balanced option for the onboard user interface. Full-color graphical output is necessary because the display must show shot placement, target information, system state, and training feedback in a way that is easy for the user to interpret. Character LCDs, passive graphic LCDs, and VFDs do not provide the graphical flexibility required for this application. OLED displays offer better image quality, but their cost becomes difficult to justify at the screen size needed for this project. Because the display should be visible from approximately 15 to 25 feet away while remaining affordable, TFT LCD is the most practical choice.

5.5.3 TFT Display Selection

After selecting TFT LCD as the preferred display technology, the next step was comparing specific TFT display candidates. The display must be large enough to fit comfortably in the upper control section of the enclosure while still providing enough resolution to show the target interface clearly. It must also communicate with the ESP32-S3 using a practical embedded interface such as SPI or I2C-supported touch control.

Touch capability is also valuable because it can simplify the physical user interface. While physical buttons will still be useful for basic controls such as mode selection, reset, and calibration, a touchscreen allows the system to support more advanced menu navigation without adding many additional buttons. This gives the project more flexibility as training modes, settings, and calibration options become more developed.

Table 5.16: TFT Display Comparison

Display	ILI9488	ST7796	AFY800480B1
Size	3.5"	4.0"	5.0"
Resolution	480×320	480×320	800×480
Interface	SPI	SPI	SPI
Cost	~$10	~$10-15	~$40
Power Draw	Low	Low	Moderate
Touch Capabilities	Yes	Yes	No

The main tradeoff is display size and resolution versus cost and integration complexity. A larger 5.0-inch display with 800×480 resolution would provide more screen area and better readability, but it would also increase cost and may require more effort to integrate cleanly. Smaller 3.5-inch or 4.0-inch SPI TFT displays are more affordable and easier to support with the ESP32-S3. Since the onboard display is mainly responsible for showing shot placement, score, mode, calibration status, and connection information, a 4.0-inch TFT display provides enough screen space while keeping the design practical.

5.5.4 Final Display Selection

The selected display is the Hosyond 4.0-inch 320 × 480 capacitive TFT touchscreen module [39], [40]. Its ST7796S display controller supports the required graphical resolution and SPI-based communication, while the integrated FT6336U capacitive-touch controller communicates through I2C [37], [38]. This arrangement allows the ESP32-S3 to control both graphical output and touch input without requiring a separate application processor.

5.6 PCB Design

5.6.1 PCB Architecture

The PrecisionShot PCB architecture is centered around separating the high-density sensor array from the control, power, and user interface sections while still keeping all subsystems electrically connected through a single organized board design. The lower target region of the PCB will contain the phototransistor array, LED feedback components, and related signal routing, while the upper control region will contain the ESP32-S3 module, display connections, buttons, power input, voltage regulation, and debugging access. This layout keeps the sensor area aligned with the physical target faceplate while allowing the control electronics to remain accessible for testing and assembly.

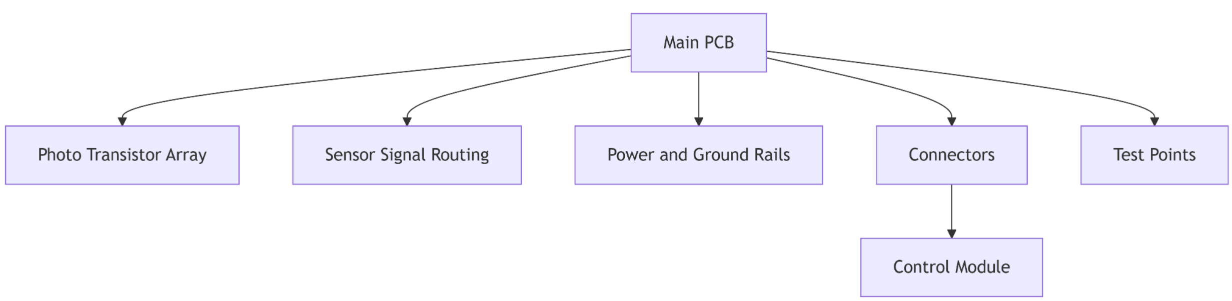


Figure 5.17: Main PCB Architecture and Subsystem Connections

5.6.2 Sensor Array PCB

The sensor array PCB houses the complete phototransistor matrix responsible for capturing and localizing incoming laser strikes across the target face. To achieve the resolution necessary for precise training telemetry, the board is configured as a dense, uniform grid of 20 columns by 25 rows, yielding exactly 500 populated phototransistor positions. Each sensor corresponds directly to a known coordinate pair on the board, allowing the central firmware to map real-time analog voltage readings directly to physical shot coordinates.

To manage the high volume of sensor lines without creating a bottleneck at the central microcontroller, the layout is organized into repeatable, modular sensor banks. These banks are routed in localized clusters that interface directly with the system's distributed multiplexer architecture, keeping individual trace runs short and balancing the signal load across the bus.

Trace routing prioritizes geometric symmetry and consistent spacing to ensure uniform signal behavior and minimize crosstalk between adjacent sensor channels.

Furthermore, component placement on the PCB is strictly bound to the mechanical dimensions of the front-facing grid plate. Every phototransistor is centered precisely beneath the 4.5 mm by 4.5 mm aperture shelf at the base of its chamfered grid cell. This precise mechanical-to-electrical alignment ensures that each sensor sits within the shadow of its respective optical well, isolating the active optical components from overlapping light paths and helping ensure that each node primarily registers strikes from its designated target location.

5.6.3 ESP32-S3 and Peripheral Connections

The control and interface section of the PCB will connect the ESP32-S3 microcontroller to the major hardware subsystems, including the sensor multiplexers, TFT display, capacitive touch controller, physical buttons, microSD card interface, battery-monitoring hardware, and optional support sensors. SPI connections will be used for higher-speed devices such as the ST7796S TFT display and microSD card, while I2C will be used for lower-speed devices such as the FT6336U touch controller, INA219 battery monitor, and optional ambient-light sensor. Digital GPIO lines will be used for multiplexer select pins, enable lines, buttons, display control signals, touch interrupts, reset functions, and other control signals.

The ESP32-S3's built-in Bluetooth Low Energy hardware will provide wireless communication with the companion mobile application and does not require a separate external wireless module or dedicated GPIO connections. The PCB will also include connections for the ESP32-S3 boot and enable/reset pins, along with native USB data lines for firmware programming, serial monitoring, and debugging. Devices sharing the SPI or I2C buses will use separate chip-select, interrupt, or reset signals where required. These connections should be clearly labeled in the schematic and PCB layout to reduce assembly mistakes and simplify testing.

5.6.4 Power Regulation PCB

The PCB power regulation system will receive a 5 V input from the selected power bank and distribute that 5 V to each component that requires 5 V, such as the screen. The voltage regulator PCB will then generate a stable 3.3 V rail for the ESP32-S3, phototransistors, multiplexers, and other low-voltage devices.

Power enters the PCB through the USB-C connector. The selected power bank provides a regulated 5 V output, which is connected to the VBUS pins of the USB-C connector. Because the USB-C connector contains multiple VBUS and ground contacts, the corresponding power and ground connections are connected together within the PCB design to distribute the input power and reduce the current carried by any individual contact. The CC1 and CC2 pins are connected to pull-down resistors to configure the connector for receiving power from the USB-C power source. The connector ground and shield connections are connected to the PCB ground system.

After entering the PCB, the 5 V input is connected to the primary power distribution rail. This rail supplies any components that require 5 V operation while also serving as the input to the 3.3 V voltage-regulation stage. Input capacitors are placed near the regulator to provide local energy storage and reduce voltage fluctuations caused by the regulator's switching operation. The placement of these components is important because the current associated with a switching converter changes rapidly during operation. Keeping the input capacitors close to the regulator reduces the length of the high-frequency current path and helps minimize unwanted noise.

The 3.3 V rail is generated using the TLV62569DBVT synchronous buck converter. Several regulator options were evaluated based on their input-voltage range, output-voltage capability, output-current capability, availability, and package characteristics. The TLV62569DBVT was ultimately selected because its input-voltage range is compatible with the regulated 5 V output of the selected power bank and it provides sufficient output-current capability for the system. The regulator is capable of supplying up to 2 A of output current and can be configured to generate the required 3.3 V output.

The LMR33620ADDA was previously considered because it supports a 3.8 V to 36 V input range and up to 2 A of output current; however, its exposed PowerPAD package would require a more complex PCB layout and thermal-via arrangement [6], [7]. This was due to the high input range of the LMR33620ADDA, which went up to 36 V, opposed to the 2.5-5.5 V input. In contrast, the selected TLV62569DBVT uses a five-pin SOT-23 package, allowing the regulator to be implemented without the same large exposed thermal pad and associated thermal-via arrangement. This simplified the PCB layout while still providing the required electrical performance for the system.

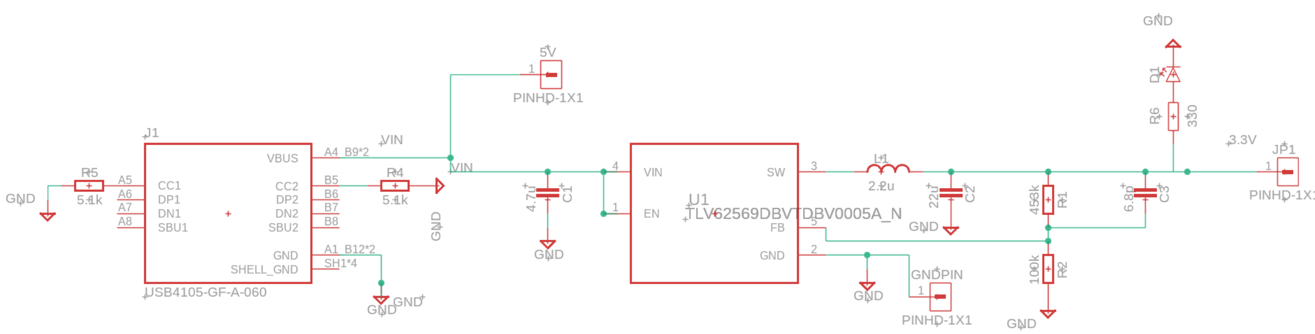


Figure 5.18: USB-C Power Input and 3.3 V Buck-Regulator Schematic

The regulator circuit converts the 5 V input into the required 3.3 V output through a synchronous buck-converter topology. The 5 V input is connected to the VIN pins of the regulator, while the enable pin is connected so that the converter is enabled when the input supply is present. The regulator's internal switching circuitry drives the SW node, which is connected to the input of the power inductor. The inductor and output capacitors filter the switching waveform and produce the regulated 3.3 V output rail.

The input capacitors are positioned between the VIN rail and ground. These capacitors provide local energy storage for the regulator and help reduce high-frequency voltage fluctuations on the input rail. The smaller ceramic capacitor is particularly important for filtering higher-frequency components of the switching current, while the larger input capacitor provides additional bulk capacitance. In the PCB layout, the input capacitors are positioned close to the regulator's VIN and ground connections to minimize the area of the input current loop.

The SW node is one of the most electrically sensitive areas of the buck-converter circuit because it carries a rapidly switching voltage. The SW connection is therefore routed only the distance necessary to connect the regulator to the inductor. The inductor then connects to the 3.3 V output rail. The output capacitors are connected between the regulated output and ground to reduce output-voltage ripple and provide a stable supply to the system components. The resulting 3.3 V rail is distributed throughout the PCB to power the ESP32-S3, sensor circuitry, multiplexer circuitry, and other 3.3 V components.

The output voltage is controlled through the feedback network connected to the regulator's FB pin. A resistor divider samples the 3.3 V output and produces a lower feedback voltage that is monitored by the regulator. The regulator adjusts its switching operation based on this feedback signal to maintain the desired output voltage. Because the feedback node is sensitive to electrical noise, the feedback components are placed near the regulator and the feedback routing is kept away from the high-current switching node whenever possible.

An output LED was added coming off the 3.3 V rail through a 10k resistor. The LED works as an indicator to show that the PCB is on and working.

The PCB layout was designed to minimize the length of high-current and high-frequency paths within the regulator circuit. The input capacitors, regulator, inductor, and output capacitors were placed in close proximity to one another to reduce parasitic resistance and inductance. The switching node was kept as compact as practical, while the feedback network was positioned away from the switching node to reduce the possibility of switching noise coupling into the feedback signal.

A ground plane is used to provide low-impedance return paths throughout the PCB. Ground connections for the input capacitors, regulator, output capacitors, and other power components are connected to the ground system. Ground vias are used where necessary to connect copper regions between PCB layers and to provide connections to the ground plane. This grounding strategy is especially important for the switching regulator because the high-frequency currents generated during operation require short and controlled return paths.

The final power-regulation PCB layout is shown in Figure 5.19. The completed design provides a regulated 5 V input distribution rail and a dedicated 3.3 V buck-converter stage. The resulting power architecture provides the voltage levels required by the system while maintaining a compact layout and reducing the potential for switching noise to interfere with the analog phototransistor measurements.

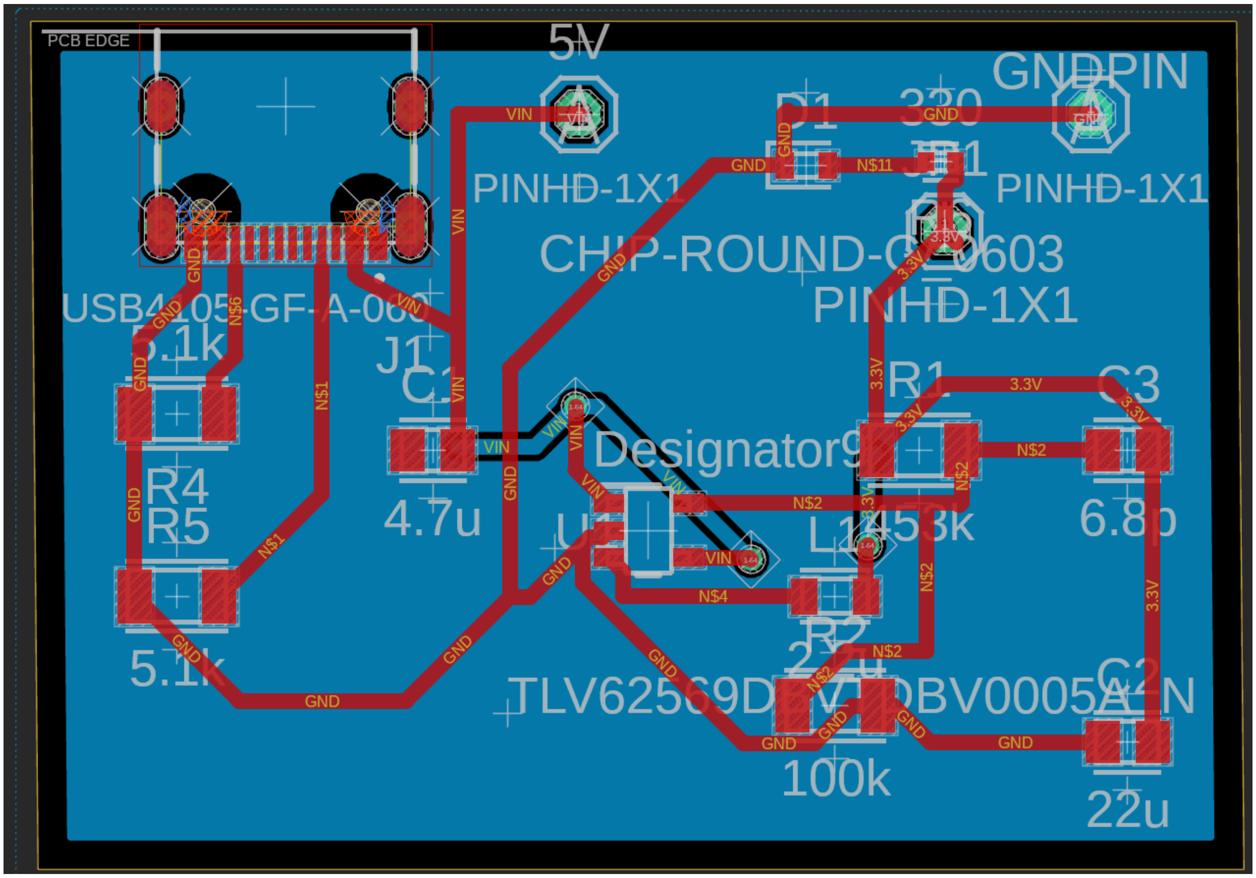


Figure 5.19: PCB Layout for the USB-C Power Input and 3.3 V Buck-Regulator Circuit

5.6.5 Debugging and Test Points

The PCB will include accessible and clearly labeled test points to support assembly, bring-up, firmware development, and troubleshooting. Power test points will be provided for the main 5 V rail, regulated 3.3 V rail, ground, regulator input and output, and the LMR33620 Power Good signal. These points will allow the team to verify voltage levels, ripple, and power stability.

Sensor-related test points will be added for selected phototransistor outputs, multiplexer outputs, channel-select lines, and enable signals. These connections will help confirm sensor response, multiplexer addressing, ADC input levels, and firmware detection thresholds.

The shared SPI bus will include test points for SCLK, MOSI, and MISO, along with the TFT and microSD chip-select signals, display data/command, reset, and backlight control. The I2C bus will include SDA and SCL, as well as touch interrupt, touch reset, and the INA219 alert signal if used.

Programming and recovery points will also be provided for USB D+, USB D−, GPIO0/BOOT, EN/RESET, 3.3 V, and ground. These connections will allow the team to use a multimeter, oscilloscope, logic analyzer, and USB Serial/JTAG tools to verify hardware behavior and compare it with firmware logs and BLE data received by the companion application.

5.7 Power System

5.7.1 Power System Overview

The PrecisionShot power system is built around a self-contained commercial semi-solid-state power bank rather than a bare lithium-polymer cell, a decision driven directly by the project's constraint against loose LiPo pouch cells and onboard lithium charging circuitry (Section 5.7.2). Using a commercial power bank as the energy source significantly simplified the power architecture compared with the team's original plan. Because the selected KUXIU power bank already contains its own internal battery management and outputs a regulated 5 V rail over USB-C, the system does not need a separate charging IC, fuel gauge, or boost converter to bring a lower battery voltage up to a usable rail. The 5 V output from the power bank is used directly as the system's primary input rail. While this power bank does feature an integrated charge indicator, the final power system will implement the INA219 power monitor to measure current across the 5V rail and provide accurate readings.

From that 5 V input, a single buck regulator steps the voltage down to the 3.3 V rail required by the ESP32-S3, phototransistor array, multiplexer network, and other low-voltage digital logic, while devices that accept 5 V directly, such as the TFT display and LED array, are powered from the 5 V rail without additional conversion. A switching buck topology was chosen over a linear regulator for this step-down stage specifically because of the efficiency and thermal advantages discussed in Section 5.7.3, and the TLV62569DBVT was ultimately selected as the specific regulator IC for this rail, as detailed in Section 5.7.4.

This two-rail architecture, an unregulated pass-through 5 V rail alongside a regulated 3.3 V rail generated by a single buck converter, is intentionally simple. It avoids the dual-regulator design considered early in the project, removes any onboard battery-charging circuitry from the PCB entirely, and keeps the power system's PCB footprint small relative to the sensor array and control sections it has to share board space with. Figure 5.20 illustrates this complete path from the power bank's USB-C output through the 3.3 V regulation stage to the rest of the system.

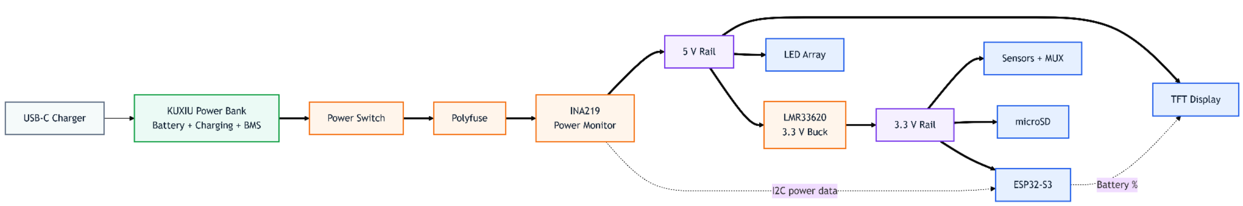


Figure 5.20: Power System Architecture and Voltage Regulation Path

5.7.2 Power Source

A commercial KUXIU 5000 mAh USB-C power bank was selected as the primary energy source for the PrecisionShot system. KUXIU markets this product line as using semi-solid-state battery technology, while independent product reviews describe its improved cycle-life and safety-oriented construction compared with conventional consumer power banks [25]–[27]. From an engineering perspective, this selection drastically minimizes the risk of thermal runaway or catastrophic puncture failures during assembly and operational testing as suggested by our professors. The power bank supplies a total energy capacity of 18.5 Wh based on a 5000 mAh capacity profile operating at a nominal voltage of 3.7V. The system utilizes the nominal 5 volts wired USB-C power delivery output channel of the power bank as its primary source.

Based on this 18.5 Wh energy threshold, battery life is mathematically bound to the active consumption state of the main board. Under worst-case continuous peak power conditions where the system draws 2.145 W, the target yields a theoretical continuous runtime of approximately 8.62 hours. Under typical dynamic training workloads where the sensor matrix idles between target acquisitions, the real-world operational lifetime extends beyond 12 hours on a single charge cycle.

Table 5.21: Estimated Power Consumption by Major Hardware Subsystem

Component	Operating Voltage	Estimated Peak Current	Estimated Power Consumption	Source
ESP32-S3 microcontroller	3.3 V	335 mA	1.17 W	Espressif ESP32-S3 datasheet
4-inch TFT LCD module	3.3 V	250 mA	0.825 W	Display module manual
500-sensor phototransistor matrix	3.3 V	450 mA	1.485 W	Team estimate
INA219 battery monitor	3.3 V	1 mA	0.0033 W	Texas Instruments INA219 datasheet
Estimated system load	3.3 V	1.056 A	3.483 W	Calculated estimate

To satisfy safety and non-invasive integration mandates, the power bank is mounted securely into an internal 3D-printed containment bracket inside the chassis. An opening is engineered directly through the physical enclosure, allowing an external USB-C cable to route to the battery's input port. While this configuration mechanically allows pass-through charging, system replenishment is bound by specific charging limitations enforced by the integrated power management hardware. The battery hub utilizes a standard 5V/2A charging profile via a USB Type-C interface, limiting input power to a maximum of 10 W. Because the system demands significant current during active multi-metric scanning, a strict fast-charging protocol is restricted during live operation to protect the semi-solid-state cells from accelerated capacity degradation. This requires the target to ideally be powered down to achieve optimal charge times and preserve long-term battery health.

5.7.3 Voltage Regulators

Before comparing specific regulator ICs, it is important to understand why linear and switching regulator topologies behave differently, because this directly explains the power efficiency and thermal tradeoffs that greatly influenced the final regulator IC selection.

To begin with, a linear regulator, including a low-dropout regulator, or LDO, regulates voltage using a series-pass transistor that behaves as a variable resistance between the input and output rails. An internal amplifier constantly compares a sampled portion of the output voltage, to a fixed reference voltage, and adjusts the conduction of the pass transistor to ensure a constant output voltage. Because the input and output currents of a linear regulator are about equal, a linear regulator can only step voltage down, dissipating the difference between the input and output voltage as heat. In other words, the power is lost as heat. The formula for the power dissipation of a linear regulator is [95]:

$$P_{\text{diss}} = (V_{\text{in}} - V_{\text{out}})\, I_{\text{out}}$$

This is the major defining limitation of linear regulators. Its ideal efficiency is approximately Vout/Vin. This means that a larger voltage difference between the input and output results in more power being dissipated as heat rather than delivered to the load. The advantage of this simplicity is that linear regulators require few external components, usually only input and output bypass capacitors. Linear regulators also produce no inherent switching noise and can provide great transient performance, making them useful for noise-sensitive applications [95].

A switching regulator, on the other hand, uses a transistor as a high-speed on/off switch rather than as a variable resistance. In a buck, or step-down set up, the switching element rapidly connects and disconnects the input supply from an inductor, at a very high frequency. The inductor and output capacitor then store and filter the transferred energy, meanwhile, a feedback loop adjusts the switching duty cycle to maintain the desired output voltage. Because the switching element operates primarily in either a low-voltage-drop or low-current state, significantly less power is dissipated as heat in the switching transistor, when compared with the pass element of a linear regulator. This allows switching regulators to achieve high efficiencies across a range of input and output voltages, with efficiencies commonly exceeding 90% depending on the specific device and operating conditions [96].

The main tradeoff is increased circuit and layout complexity. A buck converter requires an inductor, input and output capacitors, and feedback components. Additionally, the high-frequency switching operation produces voltage ripple and electromagnetic interference that must be controlled through proper filtering and proper PCB layout. The placement and routing of the switching components are therefore important considerations in the design of a switching regulator [96].

Table 5.21 estimates a total 3.3 V system load of approximately 1.056 A. With the power bank supplying a fixed 5 V rail, a linear regulator reducing this voltage to 3.3 V would dissipate:

$P_{diss} = (5\text{ V} - 3.3\text{ V}) \times 1.056\text{ A} \approx 1.8\text{ W}$

This power would be dissipated regardless of which linear regulator IC was selected. By comparison, assuming a representative switching-regulator efficiency of 90%, the power dissipated by a switching regulator would be:

$P_{out} = 3.3\text{ V} \times (1.056\text{ A}) = 3.485\text{W}$

$P_{loss} = (3.4848 / 0.90) - 3.4848 = 0.3872\text{ W} \approx 0.39\text{W}$

$1.8\text{W}/0.39\text{W} = 4.6$

This represents about a 4.6-fold reduction in power dissipated as heat compared with the linear-regulator. This difference is particularly important for the PrecisionShot system for two main reasons. First, the target enclosure is largely sealed, apart from the rear ventilation slits described, this limits the amount of airflow available to remove heat generated by the power system. Nearly 2 W of continuous heat dissipation from a single regulator would therefore create a much greater thermal load than the approximately 0.39 W expected from a switching regulator operating at the assumed efficiency.

The second main reason is that because the system is battery powered, energy dissipated as heat is energy that is not delivered to the system's electronics. By reducing the power lost during voltage conversion, the overall operating efficiency is improved, it can extend battery runtime. Together, the reduced thermal dissipation, as well as the improved power-conversion efficiency, motivated the selection of a switching buck-converter topology for the primary 3.3 V rail rather than a linear regulator. The specific regulator options considered and the final regulator selection are discussed in the following comparison.

5.7.4 Voltage Regulator Selection

The original plan for the voltage regulator system was to have two separate voltage regulators, one for 3.3V, and 5V rails that our system will require. However, after deciding to use a commercially available power bank, which includes an internal power IC that outputs a stable, and regulated, 5V, this removes the need for two separate regulator systems.

However, the 3.3V rail still leaves us with the decision of using a buck regulator, or low drop-out regulator, each having their own benefits and drawbacks for our use. The ESP32, and the phototransistor array, demand a highly stable, low-noise operating voltage. To fulfill the explicit requirement for an onboard switching regulator, the main board layout integrates a synchronous step-down, or buck, switching regulator, such as the TLV62569 or AP63203. Buck stage replaces standard linear LDOs to drastically maximize power efficiency and mitigate thermal buildup [95], [96].

Once a synchronous buck converter topology was selected, several commercially available regulators were evaluated to determine the most suitable device for the PrecisionShot Training System. Selection criteria included input voltage range, maximum output current, conversion efficiency, package style, PCB layout complexity, thermal performance, component availability, and manufacturer support.

The Texas Instruments TLV62569 was initially considered due to its excellent efficiency and compact implementation. The device accepts input voltages from 2.5 V to 5.5 V, making it well suited for single-cell lithium-ion battery applications, and is capable of supplying up to 2 A of continuous output current. Its switching frequency is approximately 1.5 MHz, which allows the use of relatively small external inductors and capacitors. However, the regulator is only available in a 2 mm × 2 mm SOT563/DSBGA package, which has extremely fine lead spacing and, in some versions, solder connections located entirely beneath the package. These package styles are intended primarily for automated assembly and are significantly more difficult to hand solder, inspect, and rework during prototype development [95].

The Diodes Incorporated AP63203 was also looked at as a potential solution. This synchronous buck regulator supports a wide 3.8 V to 32 V input voltage range, allowing considerable flexibility for future power source changes, while providing up to 3 A of continuous output current. Typical conversion efficiencies exceed 90% across much of its operating range, and the device integrates both switching MOSFETs, reducing the required external component count.

The AP63203 is available in a TSOT-26 package, which includes exposed side leads. Despite these advantages, documentation, reference layouts, and application support were less comprehensive than those provided by Texas Instruments. Since this project requires a custom PCB and extensive prototype testing, the availability of detailed layout recommendations and validated reference designs was considered an important factor in reducing design risk [96].

Older asynchronous buck regulators, including devices like the LM2596, were also considered. The LM2596 has been widely used in power supply applications and supports input voltages up to 40 V, while supplying as much as 3 A of output current. However, the regulator operates at a relatively low switching frequency of approximately 150 kHz, requiring a much larger external inductor and larger filter capacitors than more modern designs. In addition to that, the asynchronous architecture relies on an external Schottky diode, introducing additional conduction losses that reduce efficiency and increase heat generation [97].

Linear regulators such as the AMS1117-3.3 were also briefly considered because of their simplicity and extremely low component count. The AMS1117 accepts input voltages up to 15 V and can supply approximately 1 A of output current under appropriate thermal conditions. It is packaged in a readily solderable SOT-223 package, and it requires only input and output bypass capacitors to operate. However, because the device dissipates the voltage difference between the input and output as heat, supplying approximately 1 A from a 5 V input to a 3.3 V output would result in nearly 1.7 W of continuous power dissipation, as discussed earlier. This would significantly reduce battery operating time while requiring considerably more thermal management than a switching regulator. For those reasons, linear regulation was determined to be unsuitable for powering the primary system rail [98].

After evaluating the voltage-regulator requirements and the available regulator options, the TLV62569DBVT was selected for the 3.3 V power rail. The selected regulator is a synchronous buck converter capable of accepting the regulated 5 V output from the system's power bank and converting it to the 3.3 V supply required by the ESP32-S3 and other system components. The regulator provides up to 2 A of output current, providing sufficient capacity for the expected current requirements of the system while maintaining additional margin for transient loads.

The TLV62569DBVT was selected over the previously considered LMR33620ADDA primarily because it provides a simpler implementation for the system's regulated 5 V input. Since the power bank supplies a regulated 5 V output, the TLV62569DBVT's input-voltage range is well suited for the application. Additionally, the TLV62569DBVT is provided in a compact SOT-23-5 package without the large exposed PowerPAD required by the LMR33620ADDA. Eliminating the exposed PowerPAD simplifies the PCB layout by removing the need to design a large thermal pad connection and dedicated thermal-via array beneath the regulator.

The regulator's synchronous buck topology also provides a more efficient method of producing the 3.3 V rail than a linear regulator. The TLV62569DBVT uses integrated high-side and low-side switching transistors, which reduces the number of external components required and also improves power efficiency. The final regulator circuit includes the required input and output capacitors, an inductor, and feedback components specified by the regulator's design requirements. These components were placed as close as possible to the regulator to minimize the length of high-frequency current paths and reduce switching noise.

The final implementation of the TLV62569DBVT is shown in Table 5.22. The 5 V output from the power bank is supplied to the regulator's input, and the regulator produces the 3.3 V output rail used throughout the system.

Table 5.22: Voltage Regulator Options for the 3.3 V System Rail

Regulator Option	Regulator Type	Input Voltage	Output Voltage	Output Current	DigiKey Availability	Package	Source / Status
LM3940	Linear Regulator	4.5 V - 5.5 V	3.3 V	1 A	Yes	SOT-223-4	Texas Instruments/Eliminated
LM2596	Asynchronous Buck	4.5 V - 40 V	1.2 V - 37 V	3 A	Yes	TO-220-5	Texas Instruments/Eliminated
LMR33620	Synchronous Buck	3.8 V - 36 V	1 V - 24 V	2 A	Yes	SO PowerPAD-8	Texas Instruments/Eliminated
AP63200	Synchronous Buck	3.8 V - 32 V	0.8 V - 32 V	2 A	No	SO-8EP (Exposed Pad)	Diodes Inc. /Eliminated
TLV62569DBVT	Synchronous Buck	2.5 V - 5.5 V	0.6 V - 5.5 V	2 A	Yes	SOT-23-5	Texas Instruments/Selected
MP2315	Synchronous Buck	4.5 V - 24 V	0.8 V - 21 V	3 A	Yes	SOIC-8EP	Monolithic Power Sys./ Eliminated

5.7.5 Battery Monitoring

A typical cell-voltage fuel gauge IC is structurally incompatible with our design, as the commercial power bank operates as a sealed module that obscures direct access to its internal lithium cells. While the power bank features a built-in physical charge indicator, relying entirely on this hardware component provides the central microcontroller with no direct visibility into the state-of-charge or SOC telemetry, preventing the system from displaying live power metrics on the integrated user dashboard.

To overcome this visibility constraint without altering the hardware of the power bank, the system architecture utilizes a load-side Coulomb counting methodology implemented via the Texas Instruments INA219 high-side power monitor. Positioned directly across the input 5V rail on the main PCB, the INA219 measures current flow by tracking the voltage drop across an onboard, precision low-side shunt resistor. Rather than relying on simple, volatile voltage-to-capacity lookup curves which fluctuate heavily under immediate system current spikes when the 500-sensor matrix or feedback LEDs cycle, the firmware continuously integrates these current measurements over discrete time intervals.

This mathematical integration allows the system to accurately track the exact accumulation of milliampere-hours entering and exiting the device. By calculating this real-time telemetry stream against the known 5000 mAh total capacity baseline, the system can project a highly accurate, un-shifting state-of-charge percentage directly onto the TFT display panel, providing a completely standalone battery monitoring dashboard independent of any companion software applications.

5.7.6 Power Safety and Protection Mechanisms

To mitigate electrical risks during operational deployment and replenishment cycles, the power architecture incorporates multiple redundant safety layers. Because the semi-solid-state cells are housed inside a commercial power bank module, primary battery safety is regulated by an integrated battery management system internal to the battery casing. This integrated hardware controller monitors the cell pack for overvoltage, undervoltage, and thermal thresholds, terminating the charge or discharge path if any single parameter shifts outside safe operational bounds. On the custom PCB layer, overcurrent protection is established at the primary 5V input node using a surface-mount resettable polyfuse. If a short circuit or an unexpected component failure occurs anywhere down-rail such as within the LMR33620 buck converter stage or the extensive 500-sensor matrix the sharp rise in current causes the fuse to rapidly transition into a high-resistance state. This effectively isolates the sensitive mainboard electronics from the power source before any permanent hardware damage or hazardous thermal escalation can take place.

The way the battery is physically handled and mounted is also carefully planned to prevent accidental damage. The power bank sits tightly inside a custom bracket that is 3D printed directly into the target frame, separating the battery from loose wires or sharp screws. This bracket is lined with rubber strips to create a tight friction fit, which stops the battery from sliding around and protects the semi-solid-state gel inside from bumps and drops during transport. To prevent unauthorized or accidental access to the internal electronics, the enclosure is completely sealed. Reaching the battery or the main PCB requires the removal of recessed hex screws located on the rear panel of the chassis. By leaving out any external clips or latches, the design ensures that the only part of the power system a user ever interacts with is the external USB-C charging port.

5.8 Laser System:

5.8.1 Laser Technology:

The laser is a fundamental component of the PrecisionShot system because it provides the optical signal that the phototransistor array is designed to detect. The laser produces a brief pulse of light that travels toward the target each time the trigger is pressed, or a shot is fired. The phototransistor array then detects this optical signal and converts it into electrical data that can be processed by the ESP32. The performance of this sensing process depends on the many physical properties of laser light, including its wavelength, optical power, directionality, and concentration. It is therefore important to understand these characteristics when evaluating the laser as part of the PrecisionShot system.

The term laser is an acronym for Light Amplification by Stimulated Emission of Radiation. Unlike ordinary light sources, which generally produce light through spontaneous emission over a broad range of wavelengths, a laser generates light through a controlled amplification process. The resulting light is highly directional, has a narrow spectral bandwidth, and also exhibits a high degree of spatial and temporal coherence. These characteristics make lasers particularly useful in applications where a concentrated and precisely directed optical signal is required, making them perfect for the PrecisionShot [99].

Light can be described as electromagnetic radiation, which consists of oscillating electric and magnetic fields. The wavelength of the light determines many of its observable characteristics. This includes its color in the visible portion of the electromagnetic spectrum. The frequency and wavelength of electromagnetic radiation are related by the speed of light as shown by this formula [99]:

$$c = f \lambda$$

Where cis the speed of light, f is the frequency of the electromagnetic wave, and λ (lambda) is its wavelength. Light can also be described in terms of individual packets of energy called photons. The energy of an individual photon is related to its frequency and wavelength according to the formula [94]:

$$E = hf$$
$$E= hc / \lambda$$

Where E is the photon energy and h is Planck's constant. This relationship shows that shorter wavelengths correspond to higher-energy photons, while longer wavelengths correspond to lower-energy photons. Although the energy of an individual photon is important to the physical interaction between light and matter, the total optical power of a laser is also determined by the number of photons emitted per unit of time [94].

The properties of laser light are primarily produced through the process of stimulated emission. Atoms and semiconductor materials contain electrons that can occupy specific energy states. When an electron absorbs a high enough amount of energy, it transitions to a higher energy state. This excited state is unstable, and the electron eventually returns to a lower energy state by releasing energy in the form of a photon. This process is known as emission [99], [100].

Emission can occur spontaneously, in which the photon is released without an external trigger. It can alternatively occur through stimulation by another photon. During the stimulated emission process, an incoming photon interacts with an excited electron, which causes it to transition to a lower energy state. The emitted photon has properties corresponding to those of the stimulating photon, including the same energy, frequency, direction, and phase. This process allows light to be amplified in a controlled manner and is the fundamental mechanism responsible for laser operation [99],[100].

A laser requires a mechanism for generating optical gain. This is accomplished by creating a condition known as population inversion, this is when more particles occupy an excited energy state than would normally exist under thermal equilibrium. When photons pass through the gain medium, they can stimulate excited particles to emit additional photons. These photons can then stimulate further emission, creating an amplification process. The resulting optical energy is confined within an optical cavity, where repeated interactions with the gain medium increase the intensity of the light [99].

The optical cavity is generally formed using two reflective surfaces surrounding the gain medium. One mirror is highly reflective, while the other allows a portion of the amplified light to escape. The repeated reflection of photons through the gain medium allows the optical field to build up in intensity. Once the gain provided by stimulated emission exceeds the losses within the system, laser oscillation begins to occur. A portion of the resulting optical energy exits through the partially transmitting mirror and forms the laser beam [100].

This process produces several properties that distinguish laser light from light produced by conventional sources. One of these properties is monochromaticity, which refers to the relatively narrow range of wavelengths emitted by the laser. A conventional light source such as an incandescent bulb produces a broad distribution of wavelengths. In contrast, a laser is designed to produce light concentrated around a particular wavelength. This is especially useful for PrecisionShot because the response of the phototransistor array depends on the wavelength of the incident light.

Another important property is coherence. Coherent light maintains a predictable relationship between the phase of the electromagnetic waves. The process of stimulated emission causes the photons produced by the laser to have closely related phase characteristics. This allows the emitted light to maintain a more organized wavefront than light produced through independent spontaneous emissions [99].

Laser light is also highly directional compared to other light sources. Conventional light sources emit light over a wide range of angles, requiring reflectors or lenses to concentrate the light into a particular direction. A laser naturally produces a narrow beam because of the optical cavity and the properties of the emitted radiation. Although the beam still expands as it travels, its directionality allows the laser to travel toward a specific region of the target. This property is particularly important for the PrecisionShot system because the purpose of the laser is to create a localized optical signal on the target surface [99].

The laser used in the PrecisionShot system is based on semiconductor laser technology. Semiconductor lasers operate using a p-n junction formed from semiconductor materials. The p-type region contains a relatively high concentration of holes, while the n-type region contains a relatively high concentration of electrons. When the device is forward biased, electrons and holes are injected toward the junction region. When these charge carriers recombine, energy is released in the form of photons [101].

The wavelength of the emitted photons is determined primarily by the energy difference associated with the semiconductor material. The relationship between photon energy and wavelength is shown by the following formula [101]:

$$E = hc/\lambda$$

Because of this, the physical properties of the semiconductor material influence the wavelength of light produced by the laser. By selecting an appropriate semiconductor material and device structure, the laser can be designed to emit light within a desired portion of the electromagnetic spectrum [101].

At sufficiently high injection currents, the semiconductor laser can achieve the conditions necessary for optical gain and stimulated emission. The emitted photons interact with additional electron-hole pairs in the active region, causing further photon generation. The optical cavity then reinforces the desired optical modes, producing the concentrated laser output. The current at which the optical gain becomes sufficient to sustain laser operation is known as the threshold current. Below this threshold, the device may produce spontaneous optical emission, but above the threshold, stimulated emission becomes the dominant mechanism for producing the laser beam [101].

The semiconductor structure also allows laser diodes to be manufactured in relatively small packages. This is very beneficial for PrecisionShot because the laser must be incorporated into a bullet sized package.

In addition, the wavelength of the laser is particularly important because the laser and phototransistor do not operate as independent components. The phototransistor has a wavelength-dependent sensitivity described by its spectral response. As discussed previously, the selected phototransistor is most sensitive to certain wavelengths and less sensitive to others. Consequently, the laser must emit light within a wavelength range that produces a sufficiently strong response from the phototransistor array [93].

The interaction between the laser wavelength and phototransistor response can be understood by considering the optical power received by an individual sensor. If the laser emits a wavelength near the peak of the phototransistor's spectral response, a greater proportion of the incident optical power will be converted into current. If the wavelength is farther from the peak response, the same amount of optical power might produce a smaller amount of current. Because of that, the laser wavelength directly influences the signal level available to the analog sensing system [93].

This relationship is important for PrecisionShot because the system is designed to preserve analog information from the phototransistor array. The purpose of the sensor is not simply to determine whether light is present. Instead, the output voltage of each sensor is measured and compared with the outputs of neighboring sensors. A stronger optical signal produces a larger phototransistor response, which can result in a greater voltage change in the associated sensing circuit. The ability to distinguish differences in optical intensity is therefore important to the calculation of the laser's location on the target.

The spatial characteristics of the laser beam also influence the operation of the sensor array. Ideally, the laser produces a sufficiently concentrated spot that the sensors closest to the point of impact receive a stronger optical signal than sensors farther away. This creates a distribution of sensor values that can be used to estimate the location of the laser. In other words, the laser spot does not necessarily need to illuminate only one phototransistor, rather a spot that overlaps multiple sensors can provide useful information because the relative intensity measured by neighboring sensors can be used to estimate the center of the laser spot.

For example, if the laser spot is centered directly over one sensor, that sensor may produce the highest voltage response while nearby sensors produce progressively smaller responses. If the laser is located between multiple sensors, several sensors may produce significant responses at the same time. The ESP32 can then compare these analog measurements to estimate the position of the laser. This approach allows the spatial resolution of the target to be determined by both the physical spacing of the phototransistors and the signal-processing method used to interpret their outputs.

The optical power of the laser also affects the signal-to-noise ratio of the sensing system. A laser with insufficient optical power may produce a signal that is difficult to distinguish from the effects of ambient light, dark current, and electrical noise. Increasing the optical power generally increases the signal received by the phototransistors, improving the reliability of detection. However, excessive optical power can produce other problems, including saturation of the phototransistor output as well as increased laser safety concerns, which will be discussed later. The laser must therefore provide enough optical power to produce a reliable sensor signal without using more optical power than necessary.

The laser's beam divergence also influences the size of the optical spot at the target. No practical laser beam remains perfectly parallel over an unlimited distance. As the beam travels, its diameter increases according to the divergence characteristics of the laser and its optical system. A beam with lower divergence maintains a smaller spot over a greater distance, while a beam with greater divergence produces a larger spot. The appropriate beam characteristics depend on the intended distance between the laser source and the target and the desired spatial resolution of the sensor array.

For the PrecisionShot system, the laser spot size represents a very important design tradeoff. A very small spot can provide highly localized illumination but may illuminate only a small number of sensors, meanwhile a larger spot can illuminate multiple neighboring sensors and produce a smoother spatial intensity distribution, but excessive spreading can reduce the ability of the system to determine the precise point of the laser impact. It can also decrease the maximum distance that the PrecisionShot can be used. The laser’s optical characteristics must therefore be considered together with the spacing and physical layout of the phototransistor array.

The laser is also different from a conventional continuous light source in the way that it is used by the system. The intended operation of PrecisionShot involves the laser producing a short optical event corresponding to the activation of the training system. The phototransistor array must therefore detect a transient increase in illumination rather than continuously monitor only a steady-state light source. This requires the sensing and processing system to respond quickly enough to capture the laser event and retain the resulting sensor data.

The transient nature of the laser signal also provides an opportunity to distinguish the laser from background illumination. Ambient light may create a relatively constant baseline output from the phototransistors. When the laser is activated, the resulting rapid increase in illumination produces a change in the sensor outputs. The embedded system can use the measured sensor values to distinguish the laser event from the background level. This approach is particularly useful because the phototransistors respond to a range of optical wavelengths and may detect both the laser and ambient light.

The laser therefore serves as more than simply a visible aiming indicator. It acts as the optical transmitter in the PrecisionShot sensing system. The laser generates a controlled optical signal, the target's phototransistor array receives the signal, and the analog sensing circuitry converts the optical response into electrical voltages. The ESP32 then processes these measurements to determine the spatial location of the laser signal.

The electrical input determines the operating conditions of the laser. The laser's wavelength and optical power determine the characteristics of the emitted signal. The beam divergence and target distance determine how the optical signal is distributed across the sensor array. The phototransistor's spectral response and electrical characteristics determine how efficiently the light is converted into a current, and, finally, the analog circuitry and ADC determine how accurately the resulting signal can be measured by the ESP32.

This system level relationship is the reason that laser selection cannot be performed independently of the phototransistor selection. A laser with a great optical characteristic may still be unsuitable if its wavelength produces a weak response from the selected sensor. Similarly, a highly sensitive phototransistor cannot compensate for a laser that produces an insufficient or poorly distributed optical signal. The two components must therefore be evaluated as a combined optical system.

The fundamental technology of laser operations, therefore, establishes the requirements for the specific laser used in PrecisionShot. The selected device must provide a wavelength compatible with the chosen phototransistors spectral response, provide sufficient optical power for reliable detection, appropriate beam characteristics for the target geometry, and electrical characteristics compatible with the system hardware. These requirements form the base for the component evaluation and selection process discussed in Section 5.8.2.

5.8.2 Laser Selection:

The laser cartridge is the optical source of the PrecisionShot system and is responsible for generating the signal that will be detected by the phototransistor array. Contrary to a conventional laser pointer, the laser used in this project must be compatible with a firearm training application and must produce a repeatable laser shot when the trigger is activated. The selected laser must therefore satisfy both the optical and the mechanical requirements. In particular, the wavelength must be compatible with the spectral response of the selected phototransistor, about 630nm, and the optical output must be sufficient for reliable detection, and the cartridge must be capable of operating within the mechanical and electrical constraints of the training firearm.

Several commercially available laser training cartridges were researched before selecting the final device. The products compared were the following: Strikeman Laser Cartridge, LaserHIT Laser Cartridge, AeroLasec 9 mm Dry-Fire Laser Training Cartridge, and the Mantis Pink Rhino Laser Training Cartridge. These products were selected for comparison because they represent commercially available firearm training laser systems that provide a laser pulse in response to trigger activation. Although all four products serve a similar general purpose, their published specifications, activation methods, cartridge construction, laser characteristics, and intended applications differ [104-107].

The most important selection criterion for PrecisionShot was the compatibility between the laser wavelength and the spectral response of the selected phototransistor. The phototransistor used in the PrecisionShot sensor array has its highest sensitivity in the red portion of the visible spectrum, with a spectral response peak of about 630 nm. Therefore, a red laser was preferred because it could produce a stronger electrical response from the sensor array, compared to a laser operating significantly outside the phototransistor's most sensitive wavelength range.

The next major requirement was reliable laser activation. The laser cartridge must produce a consistent optical pulse when the firearm's trigger is pulled. The resulting pulse needs to be sufficiently bright and repeatable for the phototransistor array to distinguish it from ambient light. A laser that remains continuously active would not accurately represent the intended training event and could also make it more difficult for the system to distinguish a trigger event from ambient light. The duration and activation behavior of the laser are therefore important characteristics when comparing candidate products.

To start with, the Strikeman Laser Cartridge is designed specifically for dry fire firearm training and is available in a variety of firearm calibers. This product is intended to emit a laser when the firearm’s trigger is pulled, allowing the user to practice accuracy without firing live ammunition. Strikeman also specifies an effective operating range that depends on lighting conditions, with the system primarily optimized for indoor use. The company includes that bright sunlight can make laser strike detection more difficult, however this is for use with its own camera based target system.

The Strikeman cartridge represents a strong candidate for PrecisionShot because it is designed for the same general application as the intended system. Its primary advantage is its integration into a commercially developed dry fire training platform. The cartridge is designed to work with a target system and application that detects the emitted laser. However, the publicly available product information does not provide the same level of detailed optical specifications as the selected Mantis cartridge, mainly regarding the exact laser wavelength and optical output power, which is very important. This makes it more difficult to evaluate the optical compatibility between the Strikeman laser and the selected phototransistor [104].

The LaserHIT cartridge provides another useful comparison. LaserHIT offers cartridges in multiple calibers and describes its products as laser equipped training cartridges designed to emit a short laser beam when the firearm's trigger is activated. The cartridge uses a brass construction and incorporates a red Class 3A laser. The company offers different cartridge versions with different pulse durations. The HD version is designed to produce a very short laser pulse of approximately one-thousandth of a second, while the Generic version produces a longer pulse of approximately one-twentieth of a second. The company also describes different battery configurations depending on the cartridge caliber.

The LaserHIT cartridge demonstrates the importance of laser-pulse duration in a system such as PrecisionShot. A very short pulse can provide a highly defined event and reduce the amount of time during which the sensor array must monitor the laser. A longer pulse may be easier to detect and can provide additional time for the embedded system to sample the phototransistor array. The ideal pulse duration depends on the sensor response time, ADC sampling rate, multiplexing architecture, and processing speed of the ESP32.

The LaserHIT cartridge also provides a useful example of how the laser source can influence the requirements of the sensing system. A short optical pulse requires the sensor system to acquire data quickly enough to capture the event. If the laser pulse is shorter than the time required to sample the relevant sensors, the system could miss part of the optical event. On the other hand, a longer pulse provides additional time for the multiplexer and ADC to collect sensor readings, but could trigger more phototransistors in the surrounding area [105].

Continuing, the AeroLasec 9 mm Dry-Fire Laser Training Cartridge was also considered as a candidate. Its intended purpose is similar to the other products evaluated, providing a laser based dry fire training signal within a 9 mm cartridge format. However, the publicly accessible product information available for comparison provided less detailed optical information than the Mantis Pink Rhino product information. Because wavelength, output power, and laser classification are extremely important to the PrecisionShot design, a candidate with clearly documented optical specifications provides a stronger basis for engineering selection [106].

The Mantis Pink Rhino Laser Training Cartridge was ultimately selected as the preferred laser source for PrecisionShot. The product is designed to activate when the firearm trigger is pulled and uses a bright red Class 3A laser with an optical output of less than 5 mW and a specified wavelength range of approximately 630 to 670 nm. The 9 mm version uses three LR626 batteries and is designed as a rimless laser cartridge for dry fire training.

The wavelength specification was the primary reason for selecting the Pink Rhino cartridge. The specified 630 to 670 nm range is located within the red portion of the visible spectrum and overlaps the region in which the selected phototransistor provides strong optical sensitivity. The lower end of the specified laser range is especially close to the phototransistor's approximately 630 nm peak response. This provides a strong optical match between the laser source and the sensing element [107].

This compatibility can be represented by considering the spectral response of the phototransistor as a function of wavelength. If the laser wavelength is located near the peak of the sensor's response curve, a larger photocurrent is produced for a given amount of incident optical power. If the laser operates at a wavelength where the sensor is less sensitive, the resulting electrical signal will be reduced. Since the PrecisionShot system must distinguish the laser signal from ambient light and electrical noise, selecting a laser that produces a strong response from the phototransistor improves the expected signal to noise ratio.

The Pink Rhino's optical output is also appropriate for the intended sensing application. The cartridge is specified as a Class 3A laser with an output below 5 mW. This provides sufficient optical intensity for the laser to be visible and detectable by the sensor array while providing a clearly defined classification for the safety analysis in Section 5.8.3. The optical power must still be treated as a potential eye hazard, but the documented output specification provides a basis for evaluating the safety requirements of the system.

Another advantage of the Pink Rhino cartridge is its intended activation behavior. The laser is designed to activate when the trigger is pulled, producing a laser shot that corresponds directly to the simulated firing action. This is well suited to the PrecisionShot architecture because the sensor system can use the transient optical signal to identify the occurrence of a shot. The ESP32 can then process the phototransistor readings to determine the location of the laser signal on the target.

The cartridge's design is also advantageous for repeated dry-fire use in compatible, double action firearms. The design allows the cartridge to remain in the chamber during repeated trigger operations rather than requiring the user to manually eject and reload the cartridge after every simulated shot. This supports repeated testing and training sessions.

The laser candidates were compared based on several engineering criteria. These criteria included wavelength compatibility, optical output, activation behavior, cartridge compatibility, power source, and availability of technical information. The results are summarized in Table 5.23.

Table 5.23: Dry-Fire Laser Cartridge Optical and Mechanical Comparison

Parameter	Strikeman	LaserHIT	AeroLasec	Mantis Pink Rhino
Intended use	Dry-fire training	Dry-fire training	Dry-fire training	Dry-fire training
9 mm option	Yes	Yes	Yes	Yes
Laser color	Red laser	Red laser	Product-specific information should be verified	Bright red
Wavelength	Not clearly specified in reviewed product information	Not clearly specified in reviewed product information	Not clearly specified in reviewed product information	630–670 nm
Optical output	Not clearly specified in reviewed product information	Class 3A; exact output not specified in reviewed product information	Product-specific information should be verified	<5 mW
Laser classification	Not clearly specified in reviewed product information	Class 3A	Product-specific information should be verified	Class 3A
Trigger activation	Yes	Yes	Yes	Yes
Power source	Cartridge batteries	Cartridge batteries	Cartridge batteries	3 × LR626

Cartridge design	Training cartridge	Brass cartridge; rimless designs available	9 mm training cartridge	Rimless training cartridge
Primary advantage	Integrated training ecosystem	Multiple pulse-duration options	Dedicated dry-fire cartridge	Strongly documented optical compatibility
Primary limitation for PrecisionShot	Limited publicly available optical specifications	Wavelength not clearly specified	Limited verified optical data	Requires compatibility with the selected firearm

The Mantis Pink Rhino was therefore selected because it provided the strongest combination of optical compatibility and documented technical specifications. While the other products may perform equally well or better in their intended commercial training systems, the requirements of PrecisionShot are different from those of a complete commercial laser training ecosystem. PrecisionShot requires a laser source that can be evaluated as part of a custom optical sensing system. For this reason, the ability to identify the laser's wavelength and optical output is particularly important.

The selection process can also be viewed as a comparison between the laser source and the entire sensing chain. The laser must generate an optical signal, the phototransistors must convert that signal into photocurrent, the sensing circuitry must convert the photocurrent into measurable voltage, the multiplexers must route the selected sensor signals, and the ESP32 must digitize and process the resulting voltages. The selected laser must therefore provide a signal that is not only visible but also electrically measurable after passing through every stage of the optical and electronic system.

The selection of the Pink Rhino also establishes a clear relationship between the laser-selection process and the earlier phototransistor selection process. The phototransistor was selected based on its ability to detect the expected optical signal, while the laser was selected in part because its wavelength is compatible with that sensor. This shows that the components cannot be selected independently. The optical transmitter and optical receiver form a coupled subsystem, and the performance of the complete system depends on the characteristics of both components.

An additional consideration is the distinction between the standard Pink Rhino cartridge and other laser training cartridge designs. The Mantis product line includes variants with different laser operating behaviors, including versions designed around different pulse characteristics. For the PrecisionShot project, the selected cartridge must produce a sufficiently repeatable optical event for the sensor array and embedded software to capture. The final system testing will therefore be used to verify the actual laser pulse behavior, sensor response, and required ADC sampling strategy.

Based on the comparison of the available laser training cartridges, the Mantis Pink Rhino was selected as the optical source for the PrecisionShot system. Its red laser output in the 630 to 670 nm range is well matched to the spectral response of the selected phototransistor, while its sub-5 mW Class 3A classification provides a documented basis for the safety analysis. Its trigger activated operation also produces the type of laser shot required by the phototransistor array. The final laser selection therefore represents a balance between optical performance, sensor compatibility, firearm training compatibility, and safety requirements.

The selected laser's specific safety classification and operating characteristics are discussed in greater detail in Section 5.8.3. The next stage of the design process is therefore to evaluate the potential hazards associated with the laser and establish the engineering and operational controls necessary to use it safely within the PrecisionShot system.

5.8.3 Laser Safety:

Because the PrecisionShot system uses a visible laser as its optical signal source, laser safety was also considered during the selection and design of the system. The laser is intended to be directed toward the target during normal operation, this means that the beam is intentionally accessible rather than being permanently enclosed within the target electronics. This results in the potential for direct or reflected exposure at users must be considered when establishing the operating guidelines for the system. The main concern is exposure of the eye to the laser beam, since the eye can focus visible laser radiation onto a small area of the retina. Laser induced tissue damage is mainly thermal, and the eye is usually more vulnerable to laser radiation than the skin [108], [109].

The Mantis Pink Rhino laser training cartridge selected for PrecisionShot is specified by the manufacturer as a red Class 3A laser with an output of less than 5 mW and a wavelength range of approximately 630 to 670 nm. The FDA's current classification terminology identifies the corresponding low-power category as Class IIIa/3R, which depends on the classification system used. The FDA guidance describes Class IIIa lasers as potentially hazardous when directly viewed, with the risk increasing when optical aids are used. Therefore, although the selected laser has relatively low optical power compared with industrial and research lasers, it must still be treated as a potential eye hazard [108], [109].

The main hazard associated with the PrecisionShot laser is direct intrabeam exposure. This occurs when a person's eye is positioned directly in the path of the laser. This direct exposure should never be intentionally performed by a user, regardless of the relatively low power of the selected laser unit. The laser should therefore never be aimed at a person's face or eyes [109].

The second potential hazard is what is referred to as a specular reflection. A specular reflection occurs when the laser beam strikes a smooth, shiny, or otherwise highly reflective surface and is then redirected in a concentrated direction. Examples could include polished metal, mirrors, glossy surfaces, or other materials capable of reflecting the beam with relatively little scattering. Although the beam is no longer traveling directly from the laser source, a sufficiently concentrated reflection can still cause eye damage. Because of this reason, the target and surrounding test environment should be evaluated to minimize unnecessary reflective surfaces near the intended beam path [109].

Diffuse reflections are generally less concentrated because the incident optical energy is scattered over a wider range of angles. However, the beam should still not be intentionally directed at people or animals. The FDA specifically recommends that lasers not be aimed directly at people or animals and advises against using laser products that exceed 5 mW or lack appropriate power labeling [109].

The intended operation of PrecisionShot naturally reduces some exposure risks because the laser is directed toward a designated target rather than being used as a free standing laser pointer. In other words, other people shouldn't be standing down range by the target. The target provides a defined termination point for the beam, and the user is expected to aim the laser at the target surface. However, the presence of a defined target does not eliminate the possibility of an unsafe beam path. The system must therefore be operated so that people are not positioned between the laser source and the target, and the target should be placed so that the beam cannot unintentionally continue toward an area occupied by people.

The laser's classification also influenced the selection of appropriate safety controls. The FDA identifies laser products using hazard classes ranging from Class I, which is generally considered non-hazardous under normal conditions, through Class IV, which presents significant eye, skin, and potentially fire hazards. The selected Pink Rhino laser is substantially below the power levels associated with Class IIIB and Class IV systems, but its direct beam can still be momentarily hazardous to the eye. This means that the most important controls for PrecisionShot are appropriate user behavior, beam-path control, warning information, and preventing intentional direct viewing of the beam [110].

The safety requirements for PrecisionShot can therefore be divided into three categories: engineering controls, administrative controls, and operational controls.

Engineering controls are physical or design-based measures that reduce the probability of hazardous exposure. The first engineering control is the physical orientation of the laser and target. The laser should be aligned so that its intended beam path terminates at the target. The target should be positioned in a stable manner so that it cannot unexpectedly move and redirect the beam toward the user or another person. The mechanical design should also prevent the laser from being unintentionally pointed in an unsafe direction during normal operation.

The target surface also needs to be evaluated for unnecessary specular reflection. Since the PrecisionShot system detects the laser through phototransistors, the target does not need to rely on a mirror-like surface to reflect the laser back toward the user. A diffuse or non-glossy target surface is therefore preferable because it can reduce concentrated reflections while still allowing the phototransistor array to detect the incident laser signal. This provides a useful connection between the optical sensing requirements and the safety requirements of the system.

Another engineering consideration is the electrical control of the laser. The laser should only be energized when the training system is intended to produce a simulated shot. A controlled activation method reduces the possibility of the laser remaining continuously active when it isn't in use. This is particularly important during troubleshooting and testing because a continuously active laser creates a greater opportunity for accidental exposure to the eye than a laser that produces only brief, controlled pulses.

The selected laser's sub 5 mW output also represents an engineering tradeoff between detection performance and safety. Increasing the optical output would likely increase the signal available to the phototransistor array. However, increasing optical power also increases the potential hazard associated with direct or reflected exposure. The laser was therefore not selected simply to maximize optical power. Instead, the objective was to provide sufficient optical intensity for reliable detection while avoiding unnecessary increases in the optical hazard.

The second category of controls consists of administrative controls. Users of the PrecisionShot system should receive basic instructions before operating the system. These instructions should explain that the device contains a laser, identify the location of the laser source, and state that the laser must never be intentionally aimed at a person's eyes or at another person. The user should also be instructed to operate the system only when the target is properly positioned and the surrounding area is clear [110].

Appropriate warning labels should be included on the laser-training system. The label should identify the presence of laser radiation and provide the applicable laser classification and output information supplied by the manufacturer. FDA guidance states that laser products in the applicable hazard classes require appropriate warning information identifying the laser class and output power [108].

The system also needs to include operating instructions that show and detail the specific hazards associated with the intended use of PrecisionShot. These instructions should include the following requirements:

- Never intentionally aim the laser at a person or animal.
- Never intentionally look directly into the laser beam.
- Do not position a person between the laser source and the target.
- Avoid aiming the laser at mirrors, polished metal, or other highly reflective surfaces.
- Verify that the target is properly positioned before activating the system.
- Deactivate the system when it is not being used.
- Do not modify the laser cartridge or its optical output.
- Follow the manufacturer's safety instructions for the laser cartridge.

These guidelines are consistent with the general safety recommendations provided by the FDA for consumer laser products.

The third category consists of operational controls. During normal use, the user should maintain the laser's beam path toward the designated target. The target should be positioned at a location that does not allow the beam to continue into an area where another person could be exposed. When the system is being tested or aligned, the same requirement applies. Testing should not be performed by pointing the laser toward an arbitrary surface or by attempting to observe the beam directly.

The laser should also be considered during system troubleshooting. Because the PrecisionShot system contains a large phototransistor array, testing may require the laser to be activated while the sensor readings are monitored. During this process, the operator should rely on the electrical output of the phototransistor array and the system's software rather than attempting to observe the laser beam directly. The laser can be detected electronically by the target, eliminating the need for the operator to place their eyes in the beam path.

The optical alignment of the laser is another important consideration. The laser must be aligned with the intended target area so that the optical signal can be detected by the phototransistor array. However, alignment should be performed in a manner that does not require the operator to look directly along the beam path. The laser should be aligned using the target response and electronic sensor readings whenever possible.

The use of protective eyewear requires additional consideration. Laser protective eyewear is normally selected based on the specific laser wavelength and the required optical density for the exposure conditions. However, protective eyewear should not be treated as a substitute for proper beam control. For the relatively low-power laser used in PrecisionShot, the primary safety strategy should be to prevent direct exposure through beam-path control and correct operation. OSHA guidance emphasizes that protective eyewear is used when engineering controls are insufficient to eliminate the possibility of hazardous eye exposure, particularly for higher-power laser systems [110].

For the intended PrecisionShot application, the selected laser does not require the type of laser-controlled laboratory environment normally associated with Class IIIB or Class IV lasers. The system is instead designed around a lower-power consumer laser-training device. Nevertheless, the fundamental principle remains the same: the beam path should be controlled, the laser should be used only for its intended purpose, and direct eye exposure should be avoided.

The laser safety requirements also affect the mechanical design of the overall system. The target should provide a predictable surface for the laser beam to terminate on, and the surrounding structure should not contain unnecessary highly reflective surfaces near the intended beam path. The physical layout of the target should also reduce the possibility that the laser can be accidentally aimed outside of the intended sensing area during normal operation.

The safety considerations can therefore be summarized as a balance between the optical performance required by the sensor array and the exposure risk associated with the laser. A stronger laser signal could improve the signal-to-noise ratio of the phototransistor measurements, but it would also increase the potential hazard. Conversely, reducing the laser output would reduce the optical hazard but could make the sensor signal more difficult to distinguish from ambient light and electrical noise. The selected Mantis Pink Rhino provides a compromise between these requirements by providing a visible red laser signal with an output below 5 mW, while still producing sufficient optical intensity for the intended sensor system.

The final safety requirements for the PrecisionShot laser system are therefore based on the characteristics of the selected laser and the intended operating environment. The Mantis Pink Rhino laser is a sub-5 mW red Class 3A laser-training cartridge, and its direct beam should be treated as potentially hazardous to the eye. The system should be operated with the laser directed only toward the designated target, with people and animals excluded from the beam path. Highly reflective surfaces should be avoided near the beam path, and the laser should not be modified to increase its output power. Appropriate warning labels and operating instructions should also be provided [108], [109].

These safety considerations complete the laser-system design analysis. The laser technology determines how the optical signal is produced, the laser-selection process determines which device best satisfies the optical and system requirements, and the safety analysis establishes the controls necessary for responsible operation. Together, these considerations define the laser subsystem used by PrecisionShot and its interaction with the phototransistor-based sensing array.

Chapter 6: Software Design

6.1 Software Architecture

6.1.1 Software Overview

The software for the PrecisionShot Training System is responsible for controlling the main behavior of the target, processing laser shot data, calculating scoring information, updating the physical display, handling user input, managing Bluetooth communication, and supporting the optional mobile application. The software design is split into two main parts: the embedded software running on the ESP32-S3 inside the target and the mobile application running on the user’s phone. The target itself is treated as the main system, while the phone app is treated as a companion tool for remote control, analytics, logging, and history.

The embedded software will be written in C++ using the ESP-IDF development framework in Visual Studio Code. This approach was selected because the ESP32-S3 needs to handle time-sensitive tasks such as reading the phototransistor sensor array, scanning sensor groups through multiplexers, filtering false triggers, calculating shot location, updating the display, and sending Bluetooth data. Since these tasks are close to the hardware and need to respond quickly, the firmware must be reliable and organized around the real-time behavior of the device.

The basic software loop starts when the target is powered on. The ESP32 initializes the display, buttons or rotary controls, Bluetooth communication, calibration settings, local logging system, and sensor input system. After startup, the target enters a ready state where it waits for user input or a laser shot. When a shot is detected by the phototransistor array, the software reads the triggered sensor data, applies the current threshold setting, removes weak or distant noise, calculates the estimated shot location, determines the score based on the active mode, updates the display, and logs the event. If the mobile app is connected, the target also sends the shot information to the app through Bluetooth Low Energy.

The target is designed to work fully without the mobile app. Without the app, the user can still power on the system, change modes, adjust the detection threshold, enter or select distance settings, shoot at the target, and view results on the built-in display. In this standalone mode, the target provides immediate feedback such as shot location, score, current mode, calibration value, and basic system status. This is important because the PrecisionShot system should still function as a complete training device even if the user does not have a phone connected.

The companion mobile application will be developed using React Native, Expo, and TypeScript, allowing the team to use a shared component-based codebase with Expo-supported project configuration, device testing, and Android development tools [21], [22]. The app will connect directly to the ESP32 through Bluetooth Low Energy and will not require Wi-Fi, cloud storage, user accounts, or an internet connection. The app's main purpose is to make the system easier to control from a distance and to provide features that are better suited for a phone screen. These features include viewing training history, logging shot data, changing modes, adjusting distance, changing calibration settings, displaying analytics, and showing debug information when needed.

The software design also separates normal user behavior from debugging behavior. In normal operation, the system only needs to send and store the final shot result, such as the shot location, score, mode, distance, and timestamp. In debug mode, the system can send more detailed information such as active sensor IDs, threshold values, calibration data, and raw or filtered sensor readings. This allows the team to test and tune the shot detection algorithm without overwhelming the normal user interface.

Overall, the software architecture follows a device-first design. The ESP32 handles the core training system and keeps the target usable at all times, while the mobile app adds convenience, remote control, history, analytics, and debugging tools. This split allows the system to stay reliable as a standalone target while still providing the extra features needed to make PrecisionShot feel like a complete training platform.

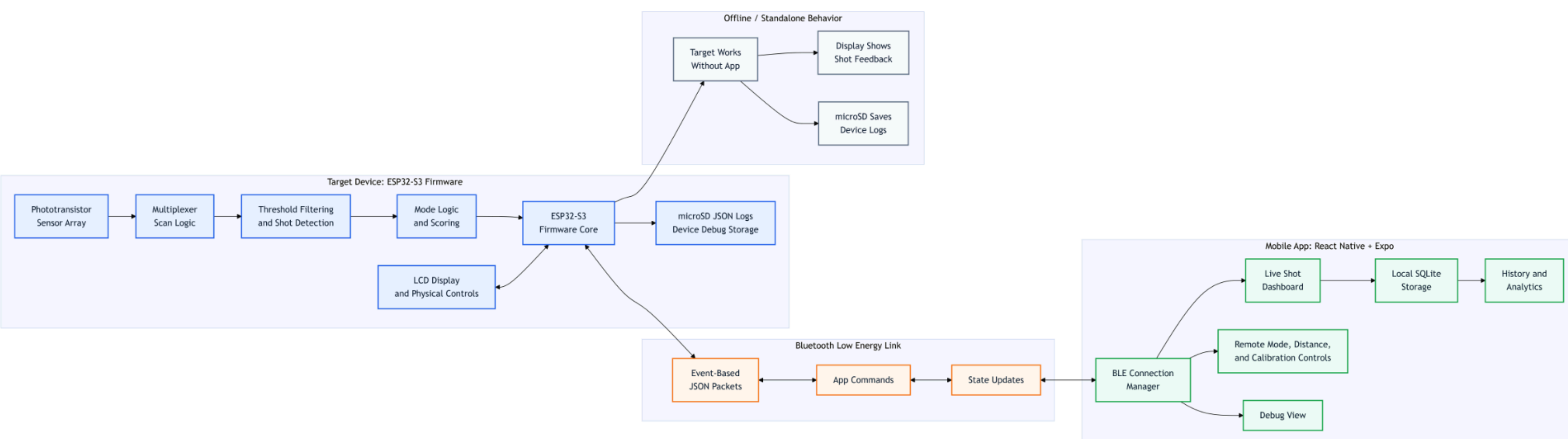


Figure 6.1: Software Overview

6.1.2 System Flowchart

The system flowchart will show the main software process from startup to shot feedback. The target begins by initializing the ESP32, display, Bluetooth system, sensor input system, local logging, calibration settings, and user controls. Once initialization is complete, the target enters a ready state and waits for either user input or a detected laser shot.

When the user changes a setting, such as mode, distance, or calibration threshold, the ESP32 updates the current system state immediately. If the app is connected, the updated state can also be sent to the app so both interfaces match. If a shot is detected, the software scans the sensor array, filters readings using the current threshold, identifies the strongest sensor cluster, rejects outliers, calculates the estimated shot location, scores the shot based on the active mode, updates the display, logs the event, and sends the data to the app if connected.

If the app is not connected, the system still completes the same target-side process. The only difference is that live app analytics are not updated. This flow keeps the target independent and ensures the main training behavior does not depend on a phone connection.

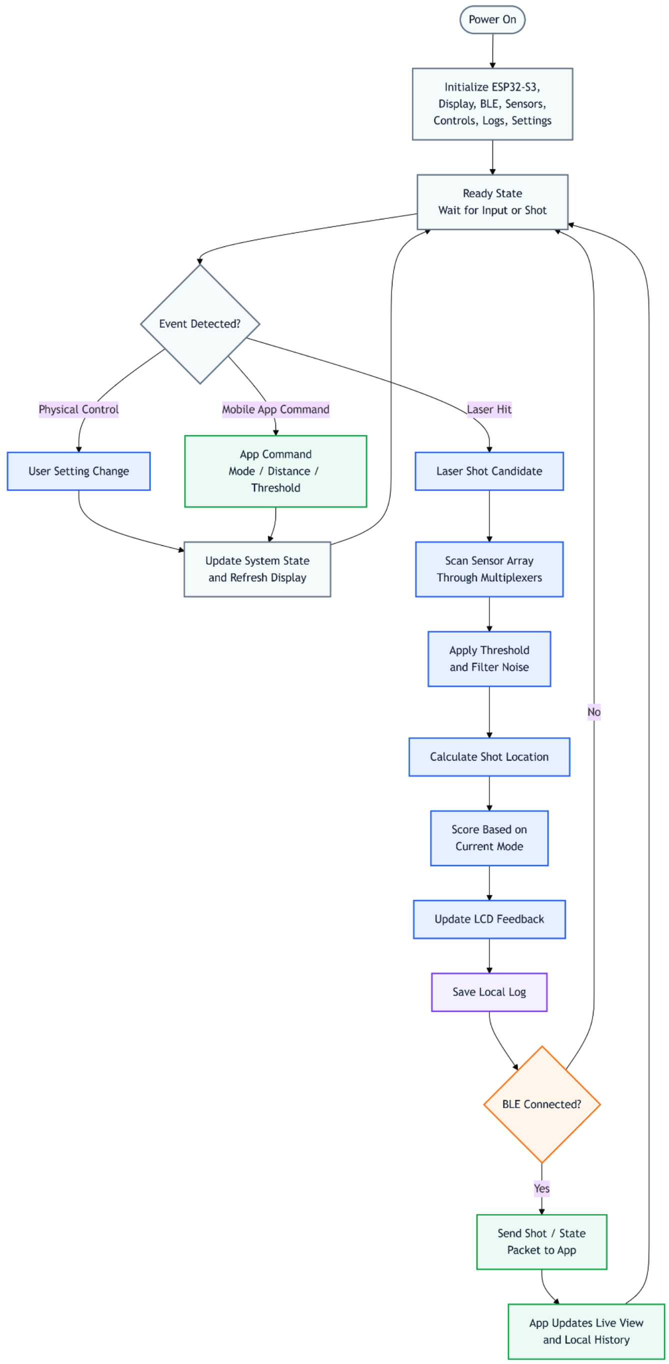


Figure 6.2: System Flowchart

6.1.3 Software Development Environment

The ESP32-S3 firmware will be developed in C++ using ESP-IDF in Visual Studio Code. This matches the embedded nature of the project because the ESP32 must directly control sensors, display output, physical controls, Bluetooth communication, logging, and timing-sensitive shot detection behavior.

ESP-IDF is preferred over the Arduino IDE because it gives the team more control over the ESP32 system and is better suited for a larger embedded project. Visual Studio Code also provides a modern development environment for organizing source files, debugging, version control, and maintaining a larger firmware codebase. During early prototype setup, publicly available tutorials covering ESP32 fundamentals, ESP32-S3 programming, development-board configuration, display integration, and touchscreen operation were consulted as supplementary learning resources [28]–[32].

The mobile app will be developed using React Native with Expo and TypeScript. This allows the team to build a mobile app with a modern JavaScript and TypeScript workflow while still targeting Android for the prototype. The app will focus on Bluetooth communication, local storage, analytics, history, and a clean user interface.

Python may still be used during development, but not as the main firmware language on the ESP32. It can be useful for testing algorithms, analyzing logs, plotting sensor data, or simulating shot behavior on a computer. The actual embedded firmware on the target will remain C++.

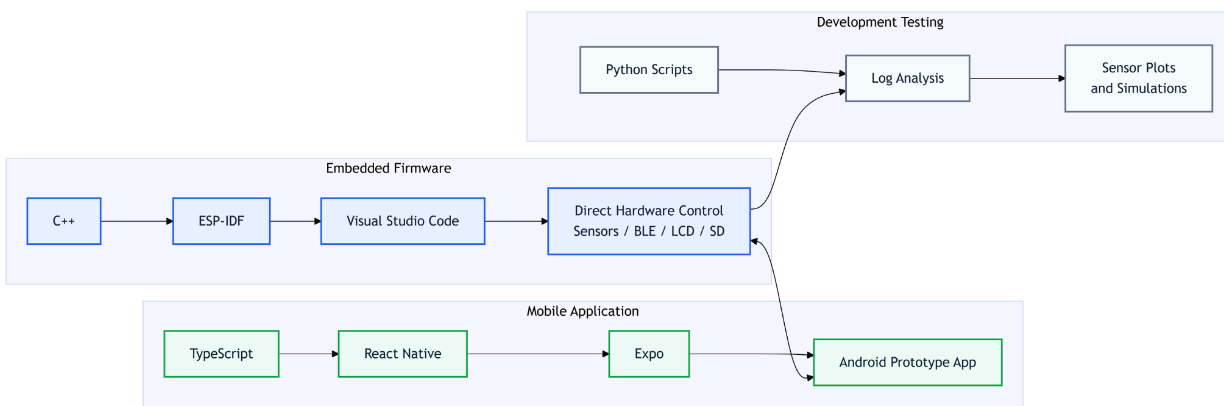


Figure 6.3: Software Development Environment

6.2 Embedded Firmware Design

6.2.1 Microcontroller Software Responsibilities

The ESP32-S3 serves as the main control unit for the PrecisionShot target and is responsible for the system's real-time embedded behavior. It will run firmware written in C++ using the ESP-IDF development framework. Because the target is designed to operate independently of the mobile application, all essential training functions will be handled directly by the microcontroller.

One of the ESP32-S3's primary responsibilities is reading the 500-phototransistor array arranged as 20 columns by 25 rows. The array will be scanned through a timer-controlled multiplexer network. At each timer interval, the firmware will select the next sensor channel, allow the analog signal to settle, read the sensor value, and store the result. After all 500 sensors have been sampled, the completed sensor frame will be analyzed for evidence of a laser strike.

Each sensor will have a unique identification number and a known physical position within the target grid. During calibration, the ESP32-S3 will record an individual ambient-light baseline for every sensor. During normal operation, each reading will be compared with its stored baseline and the system's global sensitivity setting. The firmware will reject normal ambient-light changes and weak noise while identifying sensors that show a valid increase caused by the training laser.

When a possible shot is detected, the microcontroller will identify the strongest sensor response and examine nearby sensors to determine whether they form a valid shot cluster. Neighboring readings will be used to improve the estimated impact location, while distant or unrelated readings will be rejected as noise or reflections. The firmware will also use timing and shot-lockout logic to prevent one laser pulse from being recorded multiple times across consecutive sensor scans.

The ESP32-S3 will also manage scoring and training-mode behavior. The first prototype will support Freestyle, Classic, and Rapid modes. Freestyle mode will provide unrestricted shot-location and score feedback. Classic mode will track a fixed 10-shot session, including total score and average score. Rapid mode will evaluate both accuracy and the time required to hit selected target areas. The microcontroller will maintain the current mode, score, shot count, distance setting, calibration sensitivity, and other system-state information required during a training session.

Another major responsibility is controlling the target's physical user interface. The firmware will update the onboard LCD with information such as the current mode, latest shot location, score, shot count, distance setting, calibration sensitivity, Bluetooth connection status, estimated battery status, and basic error messages. Physical controls will allow the user to change modes, adjust settings, reset the current session, and begin a new lighting calibration without requiring a connected phone.

The microcontroller will also manage Bluetooth Low Energy communication with the companion mobile application. When connected, the ESP32-S3 will send event-based JSON packets containing shot results, system-state updates, mode changes, calibration changes, and selected debugging information. The application will also be able to send commands to the target, including mode selection, distance updates, sensitivity adjustments, session resets, and debug-mode control. Changes made through the app will update the target's active system state immediately.

Local data logging will be handled through the microSD card. The ESP32-S3 will store shot records, system events, and development logs in JSON format. These records may include the estimated shot location, score, mode, distance, calibration information, sensor activity, and processing timing. Device-side storage will support troubleshooting and backup logging, while the mobile application will remain the primary location for user-facing training history and analytics.

The ESP32-S3 must also manage system reliability. The firmware will track normal operating states such as startup, calibration, ready, active training, Bluetooth connection, and debugging. It will also respond to conditions such as invalid sensor readings, multiple shot clusters, storage failures, low-power warnings, and Bluetooth disconnections. Whenever possible, the target will continue operating and display a clear message rather than allowing a single subsystem failure to stop the entire training session.

Overall, the ESP32-S3 is responsible for keeping the PrecisionShot target functional, responsive, and independent. It performs the timer-based sensor scanning, detects and locates laser shots, calculates scores, manages training modes, updates the onboard interface, communicates with the mobile app, stores local logs, and handles system errors. The mobile application expands the platform with remote controls, analytics, and training history, but the microcontroller remains responsible for all essential target operations.

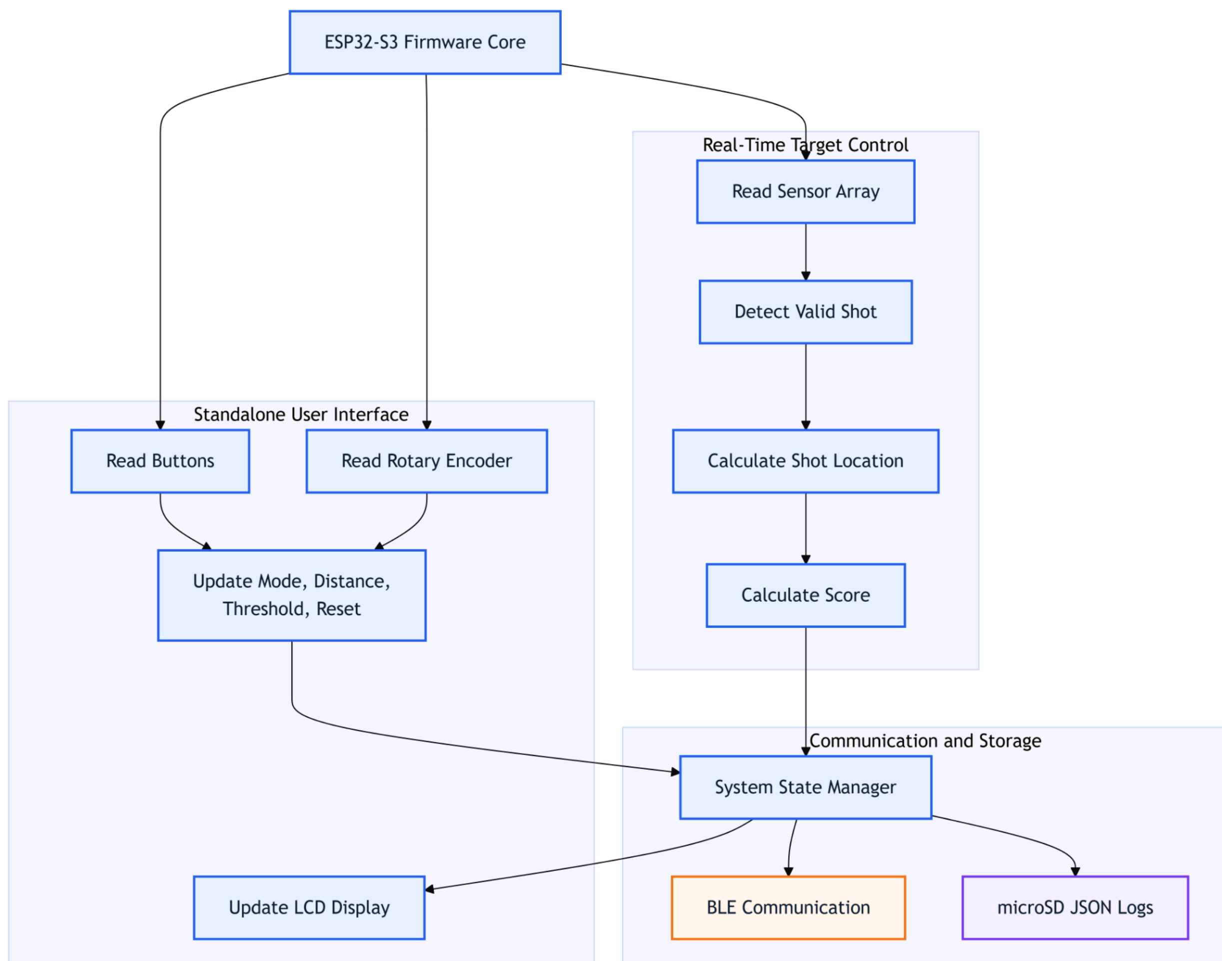


Figure 6.4: Microcontroller Responsibilities

6.2.2 Sensor Processing and Multiplexer Architecture

The PrecisionShot target uses a 500-phototransistor array arranged as 20 columns by 25 rows. Because the ESP32-S3 does not have enough analog input pins to connect directly to every sensor, the phototransistors will be read through a multiplexer network.

The sensor array will be scanned using a timer-based process. At each timer interval, the firmware will update the multiplexer address lines so that a specific sensor or group of sensors is connected to an ESP32-S3 analog input. The firmware will allow a brief settling period, read the selected sensor value, store the measurement, and then advance to the next multiplexer channel.

This process will continue until all 500 sensors have been sampled. One completed scan of the 20-by-25 array will be treated as a sensor frame. After a frame has been collected, the firmware will analyze the measurements to determine whether a laser shot occurred and where it landed.

Each phototransistor will have a unique sensor identification number and a known position within the target grid. The firmware will maintain a mapping between each multiplexer channel, sensor identification number, column position, and row position. This allows every analog reading to be associated with its correct physical location on the target.

During calibration, the firmware will scan the complete sensor array and record the normal ambient-light level of each phototransistor. This is important because sensors in different areas of the target may receive different amounts of ambient light. During normal operation, each new reading will be compared against that sensor's stored ambient baseline and the system's global sensitivity setting.

Readings that remain near their normal baseline will be ignored. Readings that increase sufficiently above their baseline will be marked as possible laser-shot candidates. The firmware will identify the strongest candidate and examine nearby sensors within the same sensor frame.

Nearby sensors with elevated readings will be grouped into a shot cluster. This allows the system to recognize that one laser strike may illuminate multiple phototransistors. Stronger readings will have more influence on the estimated shot position, while weaker nearby readings will help refine the location. Weak readings located far away from the main cluster will be rejected as noise or reflections.

If two separate groups of strong readings appear in the same frame, the firmware may treat them as separate shot candidates. Additional timing and lockout logic will prevent one laser pulse from being recorded repeatedly across several consecutive sensor frames.

The scan timer must be configured so that the complete array is read quickly enough to detect the short laser pulse. If the laser pulse is shorter than the time required to scan all 500 sensors, the hardware may require signal-hold, peak-detection, or pulse-latching circuitry so that the laser signal remains available until the corresponding sensor is sampled.

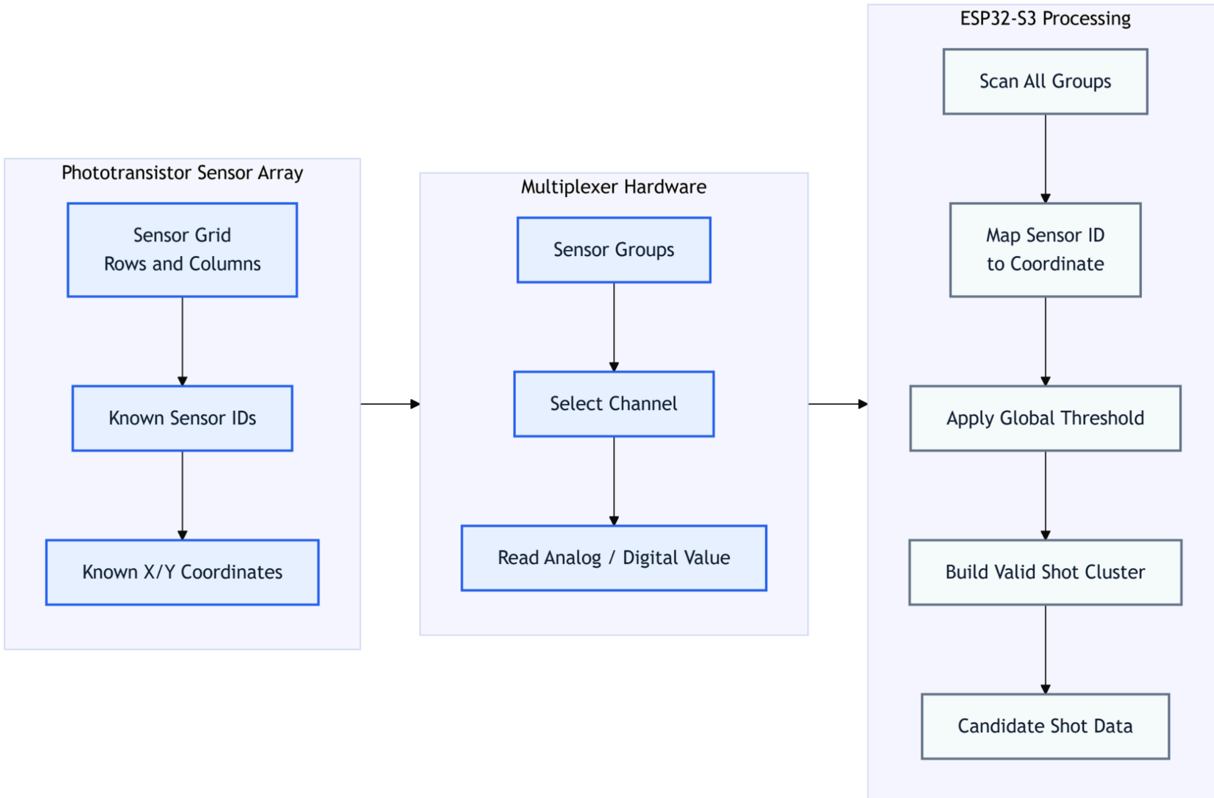


Figure 6.5: Sensor Processing and Multiplexer Architecture

6.2.3 Shot Detection Logic

The shot detection logic is responsible for determining when a valid laser shot has reached the target and estimating where the shot landed. This is one of the most important software functions within the PrecisionShot system because scoring, visual feedback, Bluetooth communication, local data logging, and mobile application analytics all depend on accurate shot detection.

The target uses a 500-phototransistor array arranged as 20 columns by 25 rows. Each phototransistor has a known sensor identification number and a known physical location within the target grid. This mapping allows the firmware to associate every sensor reading with its corresponding horizontal and vertical position.

The array will be read through the timer-based multiplexer scanning process described in Section 6.2.2. At each timer interval, the firmware selects the next multiplexer channel, allows the analog signal to settle, reads the selected sensor, and stores the result. After all 500 sensors have been sampled, the collected measurements form one complete sensor frame. The shot detection algorithm then analyzes that frame for evidence of a laser strike.

During system calibration, the firmware records the normal ambient-light level of each individual phototransistor. This allows the system to account for differences in lighting, sensor tolerances, reflections, and physical placement across the target. During normal operation, each sensor reading is compared against its own stored baseline together with the global sensitivity setting selected by the user.

The global sensitivity setting controls how much a sensor reading must increase above its normal level before it is considered a possible shot candidate. Increasing sensitivity allows the target to detect weaker laser signals, while decreasing sensitivity helps prevent false detections in brighter environments. The setting may be changed through the physical target controls or the companion mobile application.

After a sensor frame has been collected, the firmware checks whether any sensor readings increased sufficiently above their stored ambient baselines. Sensors that remain near their normal values are ignored. Sensors that show a significant increase are marked as possible shot candidates.

The firmware then identifies the sensor with the strongest response and uses it as the main reference point for the detected shot. Sensors located near the strongest response are examined to determine whether they form a valid cluster. A single laser strike may activate more than one phototransistor because the laser spot can overlap nearby cells or reflect slightly within the target's front grid structure.

Nearby candidate sensors will be grouped as part of the same shot when their positions and timing indicate that they were activated by one laser pulse. Stronger sensor responses will have greater influence on the estimated shot position, while weaker nearby responses will help refine the result. This allows the system to estimate a location between individual sensors rather than reporting only the center of the strongest sensor cell.

The shot detection process will also reject readings that do not match the expected shape of a laser strike. A weak reading located far from the primary cluster may be treated as noise, ambient-light variation, or a reflection. A separate group of strong readings located far from the first cluster may indicate that more than one shot occurred during the same scanning period.

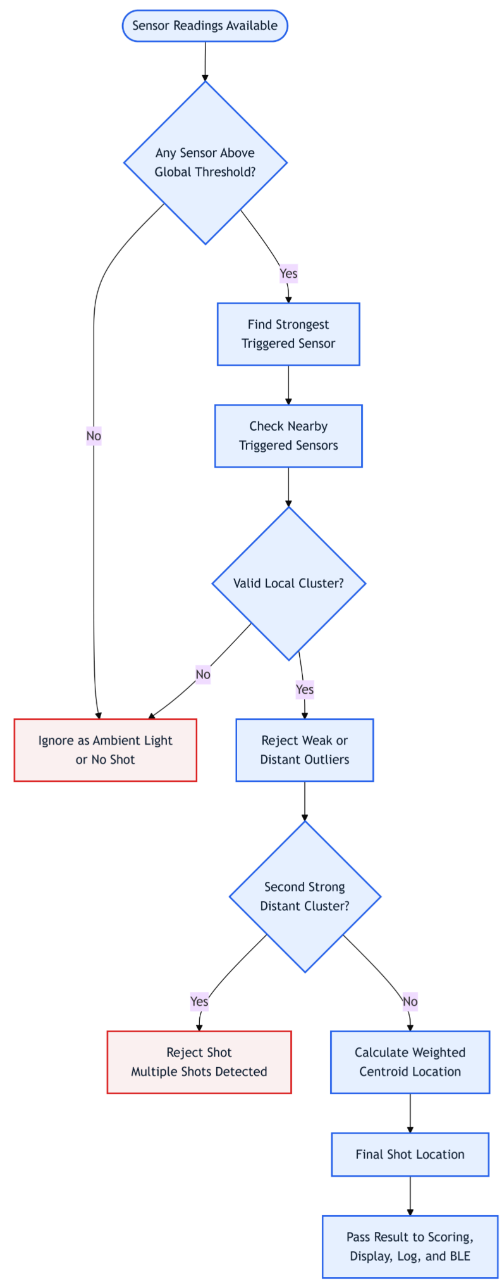


Figure 6.6: Shot Detection Logic

When multiple independent shot clusters are detected, the firmware may either process them as separate shots or reject the frame as an invalid multiple-shot event, depending on the active training mode. When rejected, the display may provide a message indicating that multiple shots were detected and that the user should repeat the attempt.

Because the array is scanned continuously, the same laser pulse may remain visible during more than one sensor frame. The firmware will therefore use shot lockout and release logic to prevent one physical laser pulse from being recorded repeatedly. After a valid shot is accepted, the system will temporarily ignore additional detections until the sensor readings return near their normal ambient levels or until a short lockout period has passed.

The timer-based scanning system must complete each array scan quickly enough to capture the laser signal. If the laser pulse is shorter than the time needed to scan all 500 sensors, the software will depend on supporting hardware such as signal-hold, peak-detection, or pulse-latching circuitry to preserve the sensor response until the corresponding multiplexer channel is sampled.

Once a valid shot location has been calculated, the firmware sends the result to the remaining software subsystems. The scoring logic determines the shot score, the display and LED systems provide immediate feedback, the logging system stores the shot record, and the Bluetooth subsystem transmits the result to the mobile application when connected.

During normal operation, only the processed shot information needs to be saved or transmitted. Debug mode may include additional information such as triggered sensor identification numbers, raw readings, ambient baselines, multiplexer channels, scan timing, cluster information, and rejected outliers. This additional data will help the development team evaluate and improve the detection algorithm during testing.

Overall, the shot detection logic does more than determine whether any sensor was activated. It collects a complete timer-based sensor frame, compares each sensor against its own calibrated baseline, identifies the strongest response, groups nearby readings, rejects unrelated signals, prevents duplicate shot records, and calculates an estimated shot location for scoring and feedback.

6.2.4 Lighting Calibration Logic

Lighting calibration is needed because phototransistors can be affected by room lighting, sunlight, reflections, component differences, and laser strength. If the detection sensitivity is too high, normal ambient light may be recorded as a shot. If the sensitivity is too low, the system may fail to detect a valid laser pulse.

During calibration, the firmware will use the timer-based multiplexer system to scan all 500 sensors in the 20-by-25 array. The normal ambient-light reading from each sensor will be stored as that sensor's baseline. Individual baselines are important because some areas of the target may naturally receive more light than others.

During normal operation, each sensor reading will be compared with its own stored baseline. A global sensitivity setting will control how much the reading must rise above that baseline before it is treated as a possible laser hit. This allows the user to adjust the sensitivity of the complete target without requiring a separate manual setting for every sensor.

The sensitivity setting will be adjustable through the target's physical controls and displayed on the onboard screen. The mobile application will also be able to change the setting through Bluetooth using options such as dark room, normal room, bright room, and custom. Any updated sensitivity value will apply immediately to future sensor scans.

The user will be able to begin a new calibration whenever the target is moved to a different location or the lighting conditions change. The system will then rescan the full array, replace the previous ambient baselines, and continue normal shot detection using the updated lighting information.

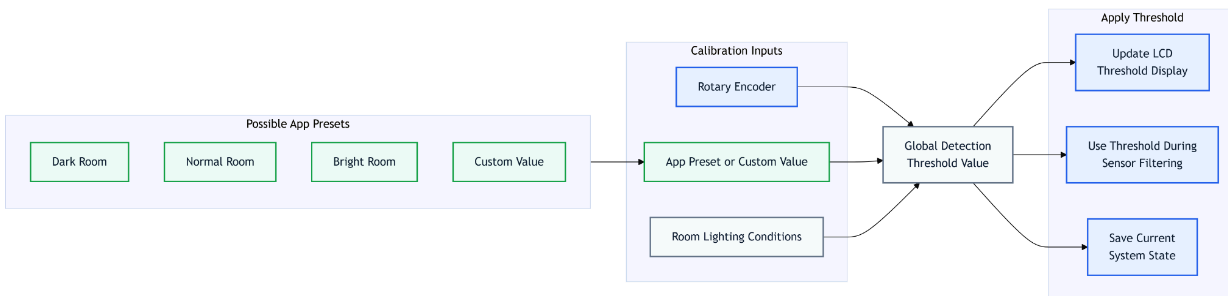


Figure 6.7: Lighting Calibration Logic

6.2.5 Scoring and Accuracy Processing

Scoring and accuracy processing will occur after the firmware calculates the estimated shot location within the 20-by-25 sensor grid. The scoring system will use the active training mode to determine how the shot should be evaluated. Because each mode supports a different type of practice, the scoring rules may change depending on whether the user is operating in Freestyle, Classic, or Rapid mode.

In Freestyle mode, the system will primarily display where the shot landed and the score associated with that position. This mode is intended for unrestricted practice, so it does not require a fixed number of shots or a formal session total. The user may continue firing until the session is manually stopped or reset.

In Classic mode, the target will use a traditional center-based scoring system. Shots closer to the center of the target will receive higher values, while shots farther away will receive lower values. The mode will record a fixed session of 10 shots and calculate the total score, average score, and overall shot grouping after the final shot.

Rapid mode will use smaller active scoring areas rather than the entire target face. The system may award more points for hitting the center of the active area and fewer points for striking near its edge. Rapid mode may also consider the time required to hit each active area, allowing both speed and accuracy to contribute to the final result.

Accuracy processing will include more than the score assigned to a single shot. The system will track the shot location, distance from the target center or active scoring area, number of recorded shots, average score, and consistency of the shot grouping. The ESP32-S3 will calculate the immediate score and update the target's display and LED feedback without depending on the mobile application.

When the mobile application is connected, it will receive the completed shot records and store them locally. The application can then calculate longer-term information such as score history, accuracy trends, performance by training distance, and performance by training mode. Moving and randomly positioned target modes are reserved as future expansion features and are not required for the first prototype.

Overall, the scoring system is designed so the target can perform all immediate scoring independently, while the mobile application provides additional history and performance analysis when connected.

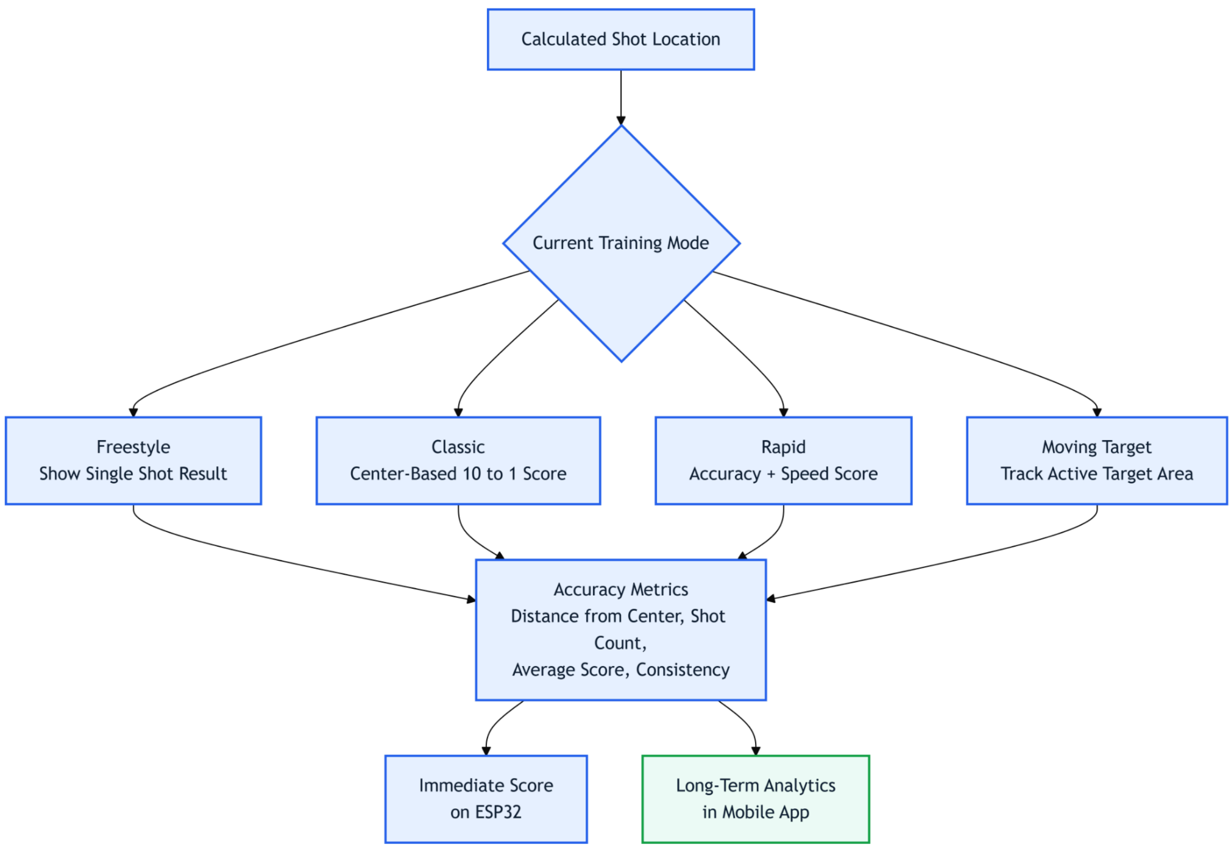


Figure 6.8: Scoring and Accuracy Processing

6.2.6 Mode Handling Logic

The target will support multiple training modes, and the software must check the active mode before processing each shot. The active mode affects how the shot is scored, what information is shown on the display, what data is sent to the mobile app, and how the target behaves after each shot.

For the first prototype, the main planned modes are Freestyle, Classic, and Rapid Mode. Freestyle mode will allow the user to shoot freely and simply see the shot location and score. Classic mode will act more like a traditional target session by tracking shot count, total score, and performance over a set number of shots. Rapid mode will focus on speed and accuracy by lighting or selecting a smaller target area for the user to hit quickly.

Modes can be changed directly on the target using the physical controls and display. The mobile app will also be able to change modes through Bluetooth when connected. Any mode change from the app or target should apply immediately so the user does not need to restart the system. When the mode changes, the ESP32 will update the current system state, refresh the display, and send the updated mode to the app if connected.

Each mode will share the same basic shot detection system, but the scoring and display behavior may change depending on the selected mode. This allows the software to reuse the same core detection logic while still giving the user different ways to train.

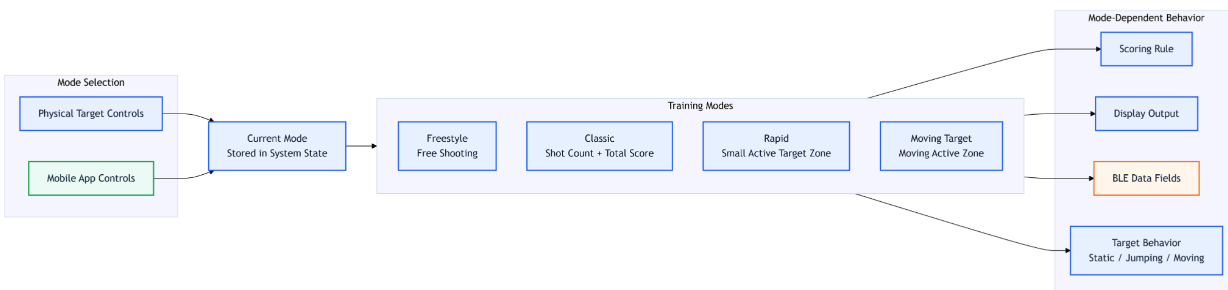


Figure 6.9: Mode Handling Logic

6.2.7 Distance Tracking Logic

Distance tracking is used to record how far the user is standing from the target during training. This is important because a shot at a close distance is not the same difficulty as a shot at a longer distance. By saving distance with shot data, the system can give the user better context for their scores, accuracy, and long-term performance.

For the first prototype, distance tracking will be handled manually instead of automatically. The user will be able to enter or select the current shooting distance through the mobile app when the app is connected. The user will also be able to set the distance directly on the target using the onboard display and physical controls. This keeps the target fully usable without the app while still allowing the app to make distance changes faster and easier from the user's shooting position.

The distance value will be stored as part of the system state. When a shot is detected, the current distance setting will be included with the shot record. If the mobile app is connected, this distance value will be sent with the shot data and saved in the app's local history. If the app is not connected, the target can still use the distance value for the display and local logging. The user will also be able to choose the preferred unit, such as feet or meters, so the system can be used in whatever measurement style is most comfortable.

Automatic distance tracking was considered as a possible future improvement, but it is not planned for the first prototype. A camera-based system or AI-based distance detection would add a large amount of extra complexity, including camera hardware, image processing, calibration, and testing. Since the main goal of the project is to build a reliable laser target system, manual distance entry is a more realistic choice for the current timeline.

In the future, the system could be expanded to support automatic distance measurement using a camera, range sensor, or other external module. For now, manual distance entry gives the project the needed distance data without making the software or hardware much harder than necessary.

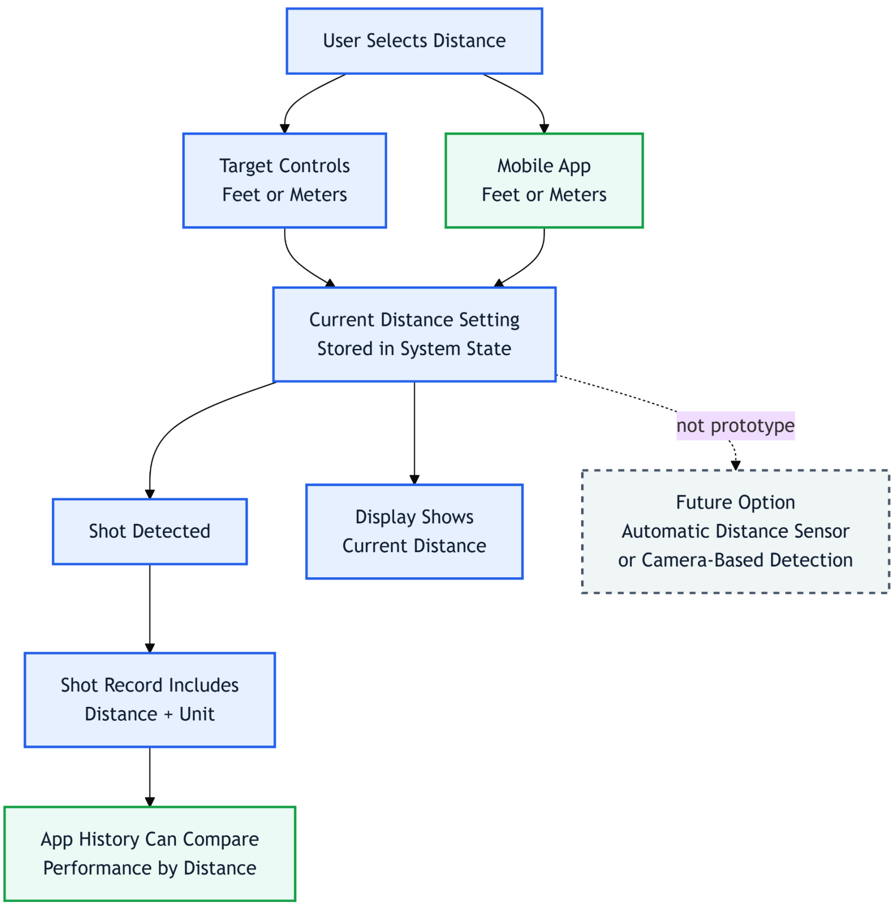


Figure 6.10: Distance Tracking Logic

6.2.8 LCD Display and Physical Interface

The PrecisionShot target will include an onboard LCD display that allows the user to view system information and operate the device without requiring the mobile application. As shown in Figure 6.11, the display will present the current training mode, latest score, estimated shot location, distance setting, calibration sensitivity, Bluetooth status, estimated battery status, and basic system errors.

During startup, the display will show the current operating mode and connection status. After a valid shot is detected, it will provide immediate feedback by showing the score and estimated impact location. If the firmware detects an invalid reading, multiple shot clusters, or another system issue, the display will provide a short message explaining the problem.

The physical interface will include buttons for changing modes, resetting the current session, and navigating system settings. A rotary encoder or similar control will allow the user to adjust the global sensitivity setting used with the calibrated sensor baselines. The displayed value will update immediately as the setting is changed.

The mobile application will provide access to the same major settings, but the target will remain fully functional when the app is disconnected. Users will still be able to select training modes, adjust the distance and calibration sensitivity, reset scores, and continue a training session directly from the target.

Overall, the onboard display and physical controls provide the primary standalone interface for PrecisionShot. The mobile application expands the system with training history and detailed analytics, while the target interface focuses on immediate feedback

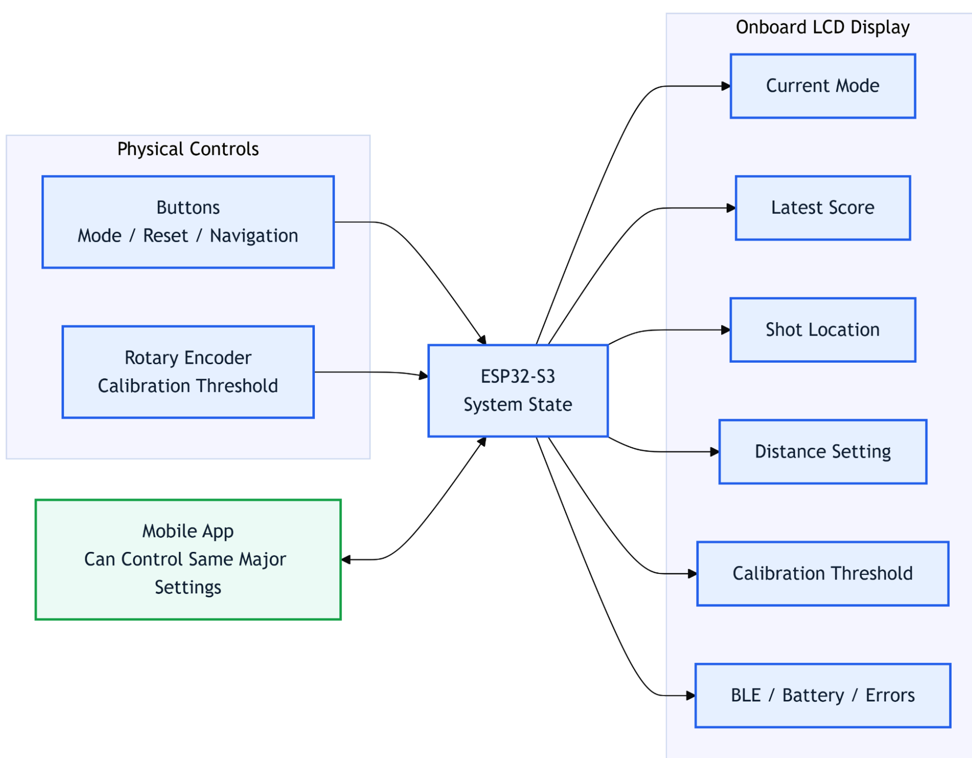


Figure 6.11: LCD Display and Physical Interface

6.3 Mobile Application Design

6.3.1 Mobile App Software Responsibilities

The mobile app is responsible for the higher-level features of the PrecisionShot system that are better handled on a phone instead of directly on the target. The target itself will still be the main device, but the app will act as a companion interface that adds remote control, training history, analytics, and debugging tools. This allows the system to remain fully usable without the app while still giving the user a more detailed and convenient way to interact with the target when the app is connected.

The mobile app will be developed using React Native with Expo and TypeScript. This stack was selected because it allows the team to build an Android app using a modern development workflow while still keeping the codebase manageable for the prototype. The Android application will connect directly to the ESP32-S3 through Bluetooth Low Energy using the scanning, connection, service-discovery, and characteristic-communication functions supported by the Android BLE framework [20]. The system will not require Wi-Fi, cloud storage, user accounts, or an internet connection. All app data will be stored locally on the phone.

One of the main responsibilities of the app is to receive shot data from the target and store it for later review. When the target detects and processes a shot, it will send a JSON data packet to the app if a Bluetooth connection is active. This packet may include the shot location, score, current mode, distance setting, timestamp, calibration value, and battery or connection status. In normal use, the app will save the final shot result. In debug mode, the app may also receive more detailed information such as active sensor IDs, threshold values, and filtered sensor readings.

The app will also provide a remote control interface for the target. Anything that can be controlled from the physical device should also be controllable from the app. This includes changing the active training mode, adjusting the manual distance value, changing the global detection threshold, resetting the current score, and enabling or disabling debug views. This is useful because the user may be standing several feet or meters away from the target and should not have to walk back to the device every time they want to change a setting.

Training history and analytics are another major responsibility of the app. The app will store shot records locally using a phone-side database, likely SQLite. For the prototype, the system will use a single local user profile, and shot history will be grouped mainly by date instead of requiring formal login accounts or cloud-based sessions. This keeps the app simple while still allowing the user to review previous training data, compare performance over time, and see how distance, mode, and accuracy affect their results.

The app interface should also handle situations where the target is not connected. If no PrecisionShot device is connected but previous history exists, the app can still show saved training history and display a disconnected status message. If there is no device connected and no previous history, the app should mainly show the connection screen and allow the user to scan for the PrecisionShot target. Once connected, the app will request the current target state and update the user interface to match the device.

Overall, the mobile app is not required for basic target operation, but it greatly improves the user experience. The target handles the real-time embedded work, while the app handles remote control, local history, analytics, and debugging. This separation keeps the PrecisionShot system reliable as a standalone training device while still giving the user a more complete training platform when the phone is connected.

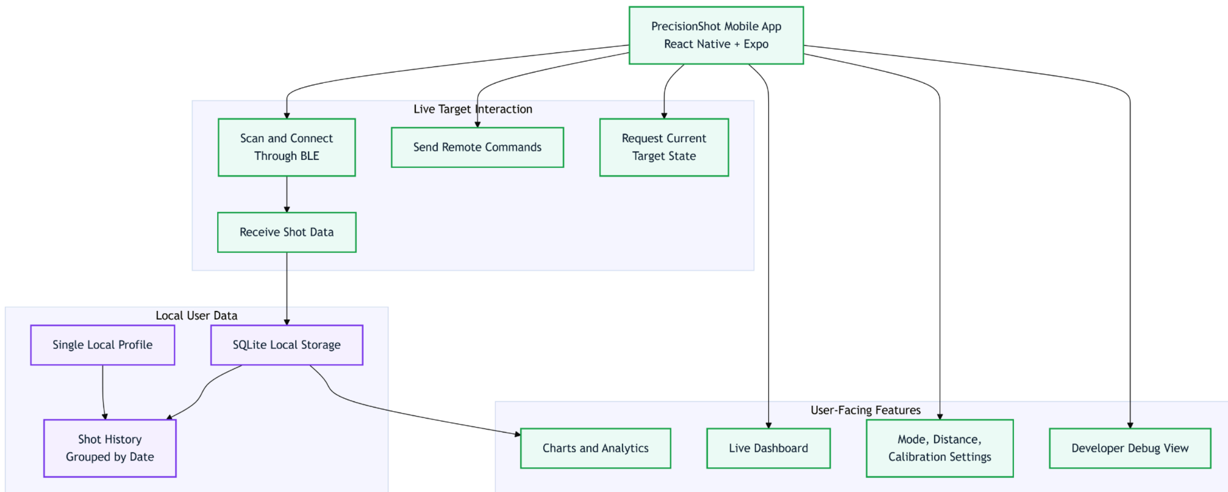


Figure 6.12: Mobile App Software Responsibilities

6.3.2 Mobile Application Architecture

The mobile application will be organized into five main layers: the user interface layer, Bluetooth communication layer, application-state layer, local database layer, and analytics layer. Separating the application into these layers will make the software easier to develop, test, maintain, and expand as additional features are added.

The user interface layer will control what the user sees and interacts with. It will include the live dashboard, Bluetooth connection screen, training history, analytics pages, mode selection, distance settings, calibration controls, and debugging views. When the target is connected, the application will display live shot information and the current state of the target. If the target is disconnected but stored history is available, the user will still be able to review previous results. If no target is connected and no history has been recorded, the application will focus on scanning for and connecting to a PrecisionShot device.

The Bluetooth communication layer will manage communication between the mobile application and the ESP32-S3. It will handle device scanning, connection establishment, disconnection detection, reconnection, incoming JSON packets, and outgoing control commands. Shot packets received from the target will be decoded and passed to the application-state layer. The Bluetooth layer will also send commands for mode changes, distance updates, calibration adjustments, session resets, and debug-mode settings.

The application-state layer will maintain the current condition of the connected target and the mobile interface. This includes the active training mode, most recent shot, score, distance setting, calibration value, battery status, Bluetooth connection state, and current session information.

Keeping this information in a shared application state will allow all screens to display consistent data and update immediately when new information is received from the target.

The local database layer will store shot records and training history on the user's phone. Each stored record may include the shot location, score, timestamp, mode, distance, calibration value, and other information needed for later review. The application will use this stored data even when the target is disconnected, allowing users to review previous training sessions without an active Bluetooth connection or internet access.

The analytics layer will process the stored shot data and calculate useful performance information. This may include average score, shot count, accuracy trends, performance by training mode, performance by distance, and shot-group consistency. These results will be displayed through summaries, charts, and history views that help the user evaluate improvement over time.

This layered architecture keeps the responsibilities of the application separated while allowing the different parts to work together. The ESP32-S3 remains responsible for real-time shot detection, scoring, and target control, while the mobile application focuses on remote control, data storage, training history, and performance analysis.

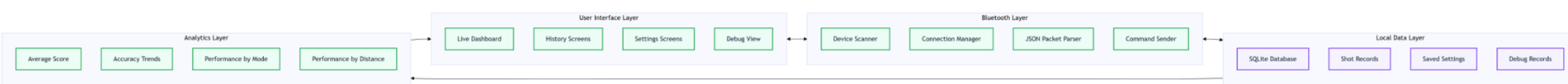


Figure 6.13: Mobile Application Architecture

6.3.3 Mobile App Interface and Analytics

The mobile app will provide a more detailed interface for reviewing training data and controlling the target from a distance. While the onboard display is used for quick feedback, the app gives the user a larger screen for viewing shot history, accuracy trends, scores, distance settings, and other training information. The app is not required for the target to function, but it makes the system easier to use and gives the user more useful feedback after training.

The app will connect directly to the PrecisionShot target through Bluetooth Low Energy. Once connected, it will receive shot data from the ESP32 and save that data locally on the phone. The app will not require cloud storage, user accounts, Wi-Fi, Firebase, or an internet connection. For the prototype, the app will use one local profile, and training history will be grouped mainly by date. This keeps the app simple while still allowing the user to review previous results.

The app interface will show live shot information when the target is connected. This may include the current mode, latest shot location, score, distance, calibration threshold, battery level, and Bluetooth connection status. The app will also allow the user to change the same settings that can be changed from the device, such as training mode, distance, calibration value, score reset, and debug mode. Any changes made in the app should apply immediately to the target.

For analytics, the app can calculate and display information such as average score, shot count, accuracy trends, performance by distance, and performance by mode. Since the phone has more storage and a better screen than the target display, it is a better place for graphs, charts, and longer-term training history. This allows the user to see whether their accuracy is improving over time instead of only seeing one shot at a time.

The app will also include a debug or developer view for testing. In this mode, the app may show extra information such as active sensor IDs, threshold values, calibration data, and raw or filtered sensor readings. This debug information will help the team test the sensor array, tune the shot detection logic, and find problems during development.

If the target is not connected, the app can still show saved local history if any exists. If there is no connected target and no saved history, the app will mainly show the Bluetooth connection screen and allow the user to scan for the PrecisionShot device. Overall, the app acts as a companion interface that adds remote control, local history, analytics, and debugging while keeping the target itself as the main training system.

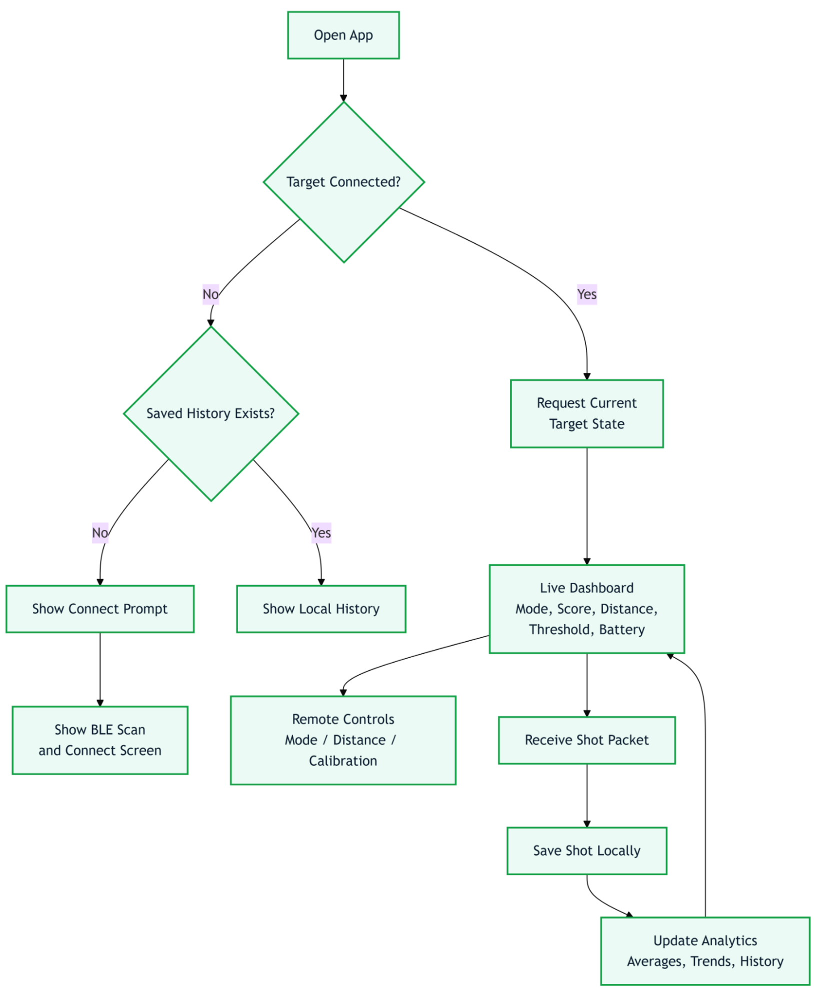


Figure 6.14: Mobile App Interface and Analytics

6.3.4 Local Database Architecture

The mobile application will store all training history locally on the user's phone. The prototype will not require cloud storage, user accounts, or an internet-based database. Keeping the data on the device improves privacy, simplifies development, and allows users to review their training history without an internet connection.

The prototype will use a single local user profile. Additional profiles may be added in a future version, but they are not required for the initial system. Training records will be organized by date and session so that shots from the same practice period or Classic-mode round remain grouped together.

The application will use SQLite as its local database. Each shot record will store information such as the timestamp, session identifier, training mode, score, horizontal and vertical shot location, selected distance, measurement unit, and calibration sensitivity. Session records may also store summary values such as total shots, average score, and session start time.

Development and debugging information will be stored separately from normal training history. This prevents raw sensor readings, timing data, and diagnostic messages from cluttering the user-facing records.

The mobile database will serve as the primary location for reviewing training history and performance analytics. The target's microSD card will provide device-side logs and backup records, while the mobile application will organize the information into a format that is easier for the user to review over time.

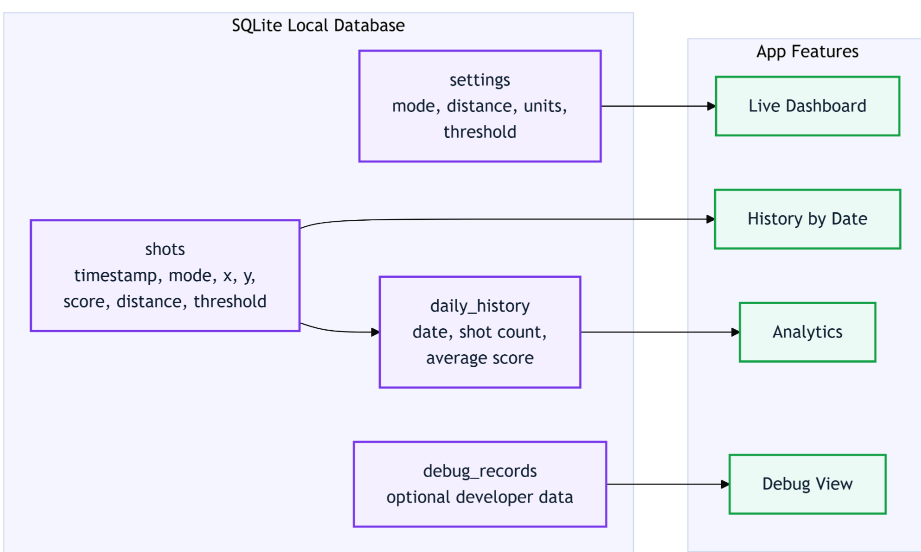


Figure 6.15: Local Database Architecture

6.4 Storage and Wireless Communication

6.4.1 Local SD Card Storage

The target may include local storage using a microSD card. The main purpose of the microSD card is not to replace the mobile app's training history, but to give the device a place to store logs, debug data, and backup shot records during testing. This is useful because the target should still be able to record basic system behavior even when the app is not connected.

For the prototype, microSD logging is planned to be always on. Each shot event can be saved as a JSON record containing information such as the shot location, score, active mode, timestamp, distance setting, calibration threshold, and Bluetooth connection state. In debug mode, the log may include more detailed information such as active sensor IDs, sensor values, threshold data, and processing timing.

Using JSON makes the logs easier to read and debug during development. Since the data may include nested information, such as multiple triggered sensors for one shot, JSON is easier to work with than a basic table format. This also allows the team to review logs later on a computer and compare the recorded data against physical testing results.

The mobile app will still be the main place for user-facing history and analytics. The microSD card is mainly for device-side records, troubleshooting, and development testing. If the stored logs become too large, the firmware may later include an automatic cleanup process that deletes older log files or clears records when the card is close to full.

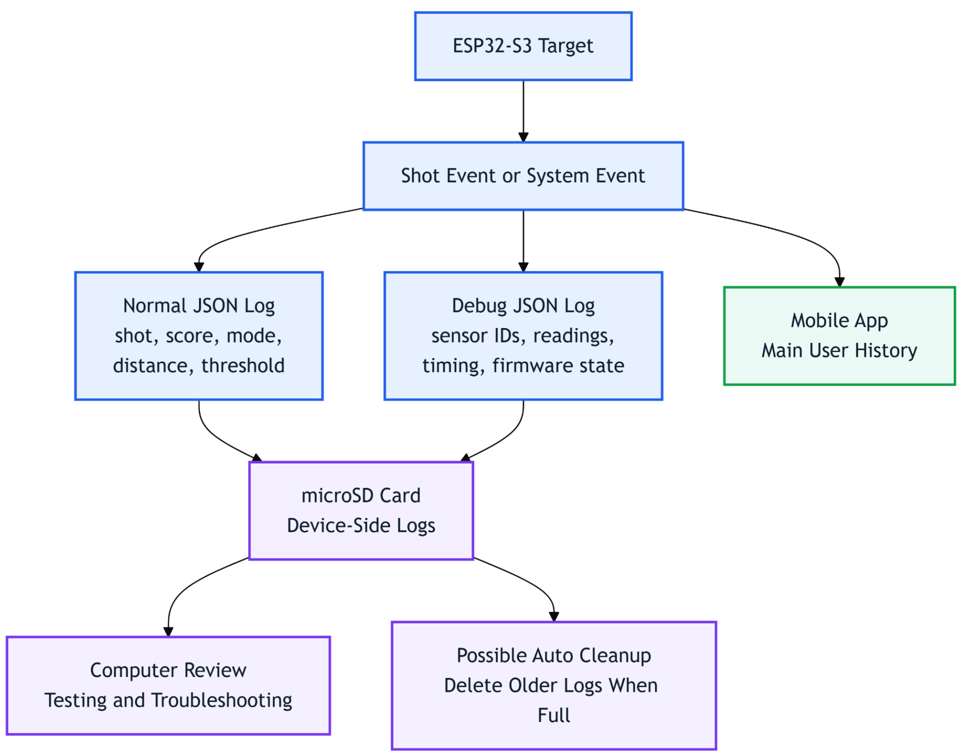


Figure 6.16: Local SD Card Storage

6.4.2 Bluetooth Connection Management

Bluetooth connection management controls how the mobile app connects to the PrecisionShot target. The ESP32-S3 will advertise itself as a Bluetooth Low Energy device, and the app will scan for nearby PrecisionShot targets. For the prototype, the system will allow one active phone connection at a time.

The connection process should be simple for the user. The user opens the app, scans for the PrecisionShot device, selects the correct target, and connects. The app should not require a cloud account or internet connection. Once connected, the app can request the current target state and display the correct mode, distance, calibration value, battery status, and connection status.

If the Bluetooth connection is lost, the target will continue operating normally. The app will stop receiving live data and show a disconnected status, but the target will still detect shots, calculate scores, update the display, and log locally if microSD storage is available. This supports the device-first design of the system.

When the app reconnects, it can request the current system state from the target. This allows the app to update its interface without assuming that the target is still in the same mode or settings as before the disconnection.

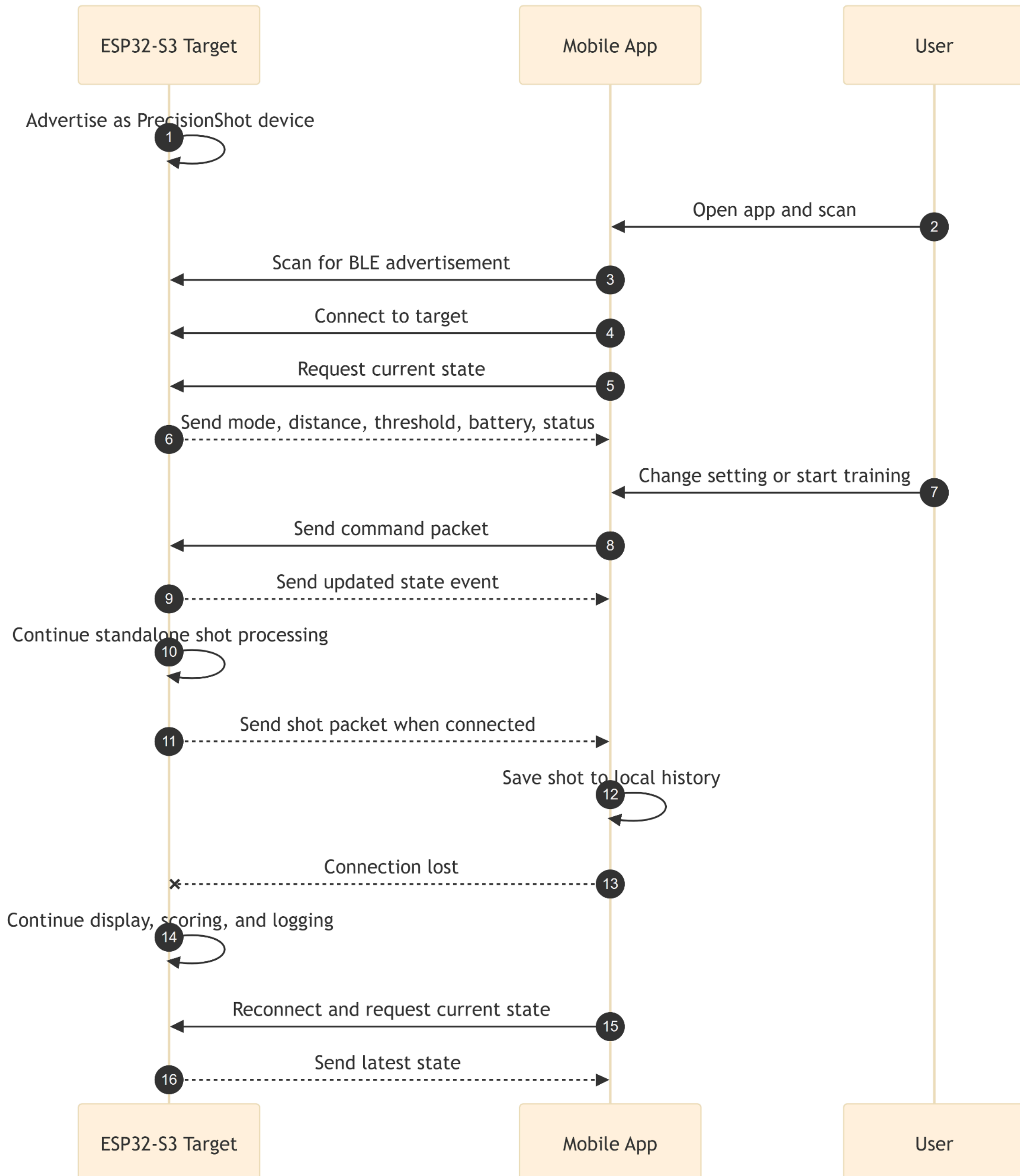


Figure 6.17: Bluetooth Connection Management

6.4.3 Communication Protocol Design

The PrecisionShot communication protocol is designed around Bluetooth Low Energy communication between the ESP32-S3 and the mobile app. The target will advertise itself as a PrecisionShot device, such as PrecisionShot-339, and the app will scan for and connect directly to it. This avoids the need for Wi-Fi, cloud services, user accounts, or an internet connection.

The protocol will be event-based instead of constantly streaming data. This means the ESP32 will only send data when something important happens, such as a shot being detected, a mode changing, a calibration value changing, or a debug event occurring. This keeps the communication simple and avoids unnecessary Bluetooth traffic.

The app will also be able to send commands back to the target. These commands may change the training mode, update the distance setting, adjust calibration, reset the current score, or enable debug mode. Commands will apply immediately so the app can act like a remote version of the physical controls on the target.

For the prototype, the system does not require a separate acknowledgment packet for every command. Instead, the target will update its state and send state changes back to the app when needed. This keeps the protocol simpler while still allowing the app to confirm the current target state through regular state updates.

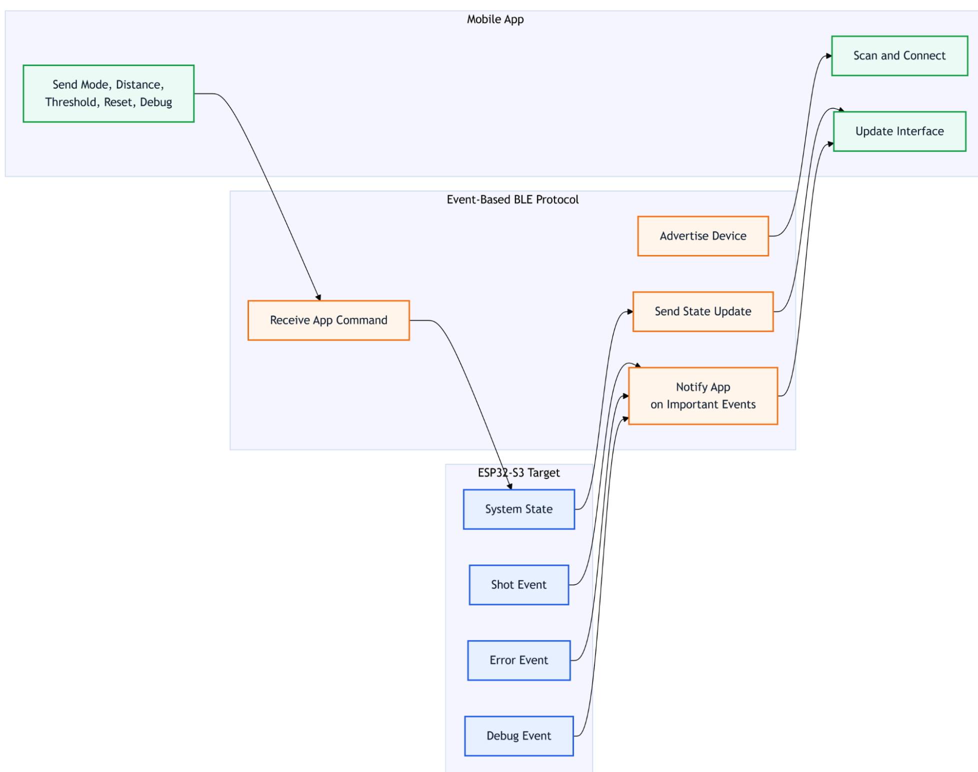


Figure 6.18: Communication Protocol Design

6.4.4 Data Packet Structure

The first prototype will use JSON data packets for Bluetooth communication. Although JSON is larger than a binary format, it is easier to read, test, and debug during development.

A normal shot packet will contain the processed shot result, including the shot number, x and y coordinates, score, mode, timestamp, distance, calibration threshold, and battery level. An example packet is shown below:

```
{
 "shotNumber": 7,
 "x": 12.4,
 "y": 8.9,
 "score": 9,
 "mode": "Classic",
 "timestamp": "2026-07-27T14:32:18",
 "distance": 10,
 "distanceUnit": "ft",
 "calibrationThreshold": 120,
 "batteryLevel": 82
}
```

Debug packets may also include sensor IDs, raw or filtered sensor values, timing information, and firmware state. Full sensor-array data will only be transmitted when requested to avoid unnecessarily large Bluetooth packets.

If JSON becomes too slow or too large during testing, the system may later switch to a compact binary format.

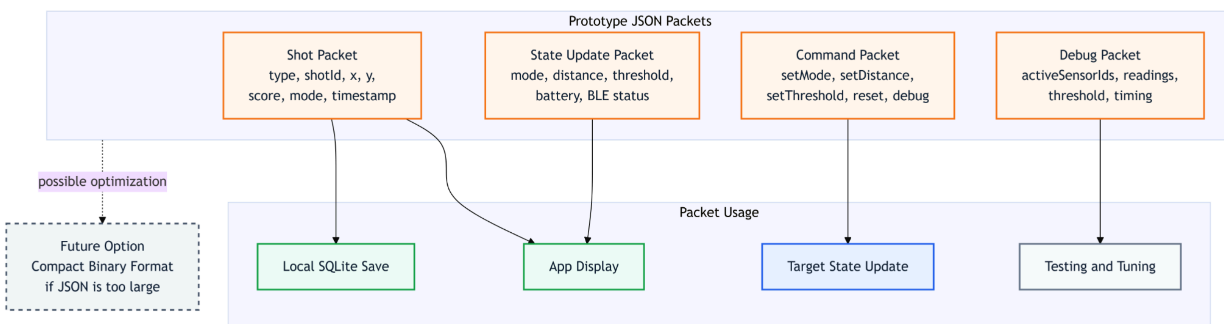


Figure 6.19: Data Packet Structure

6.4.5 Wireless Data Flow

The PrecisionShot system will use Bluetooth Low Energy to send data between the target and the mobile app. The phone will connect directly to the ESP32-S3 inside the target, so the system will not require Wi-Fi, cloud services, user accounts, or an internet connection. The Bluetooth connection is mainly used for sending shot data to the app and allowing the app to control the same settings that can be controlled from the target itself.

The target will use an event-based data flow. This means that the ESP32 will not constantly send data when nothing is happening. Instead, wireless packets will be sent when an important event occurs, such as a shot being detected, the mode changing, the distance value changing, the calibration threshold changing, debug mode being enabled, or an error state occurring. This keeps the wireless communication simpler and reduces unnecessary Bluetooth traffic.

For the first prototype, the wireless data packets will use JSON. JSON is not the smallest or fastest format, but it is easy to read, test, and debug. This is helpful during development because the team can quickly see what data is being sent between the ESP32 and the mobile app. If Bluetooth packet size or speed becomes a problem later, the system may be changed to use a more compact binary format.

In normal operation, the target will send the final processed shot result to the app. This may include the shot location, score, active mode, timestamp, distance setting, calibration threshold, and battery or connection status. In debug mode, the target may send larger packets that include extra information such as active sensor IDs, sensor values, threshold information, and processing timing. Full sensor-array dumps should only be sent when requested for debugging because sending too much sensor data every shot could make Bluetooth communication slower.

The app will also be able to send commands to the target. These commands may include changing the mode, updating the distance, adjusting the calibration threshold, resetting the current score, or enabling debug mode. These commands will apply immediately so the app can act as a remote control for the target. The system does not plan to use a separate acknowledgment packet for every command. Instead, the target will update its current state and send state changes back to the app when needed.

If the Bluetooth connection is lost, the target will continue operating normally. Shot detection, scoring, display updates, mode handling, and local logging will continue on the device. The app will simply stop receiving live data and show that the target is disconnected. When the app reconnects, it can request the current system state from the target and continue receiving new shot data.

Overall, the wireless data flow is designed to support the app without making the app required for basic use. The target remains the main system, while Bluetooth allows the phone to receive data, control settings, display analytics, and help with debugging.

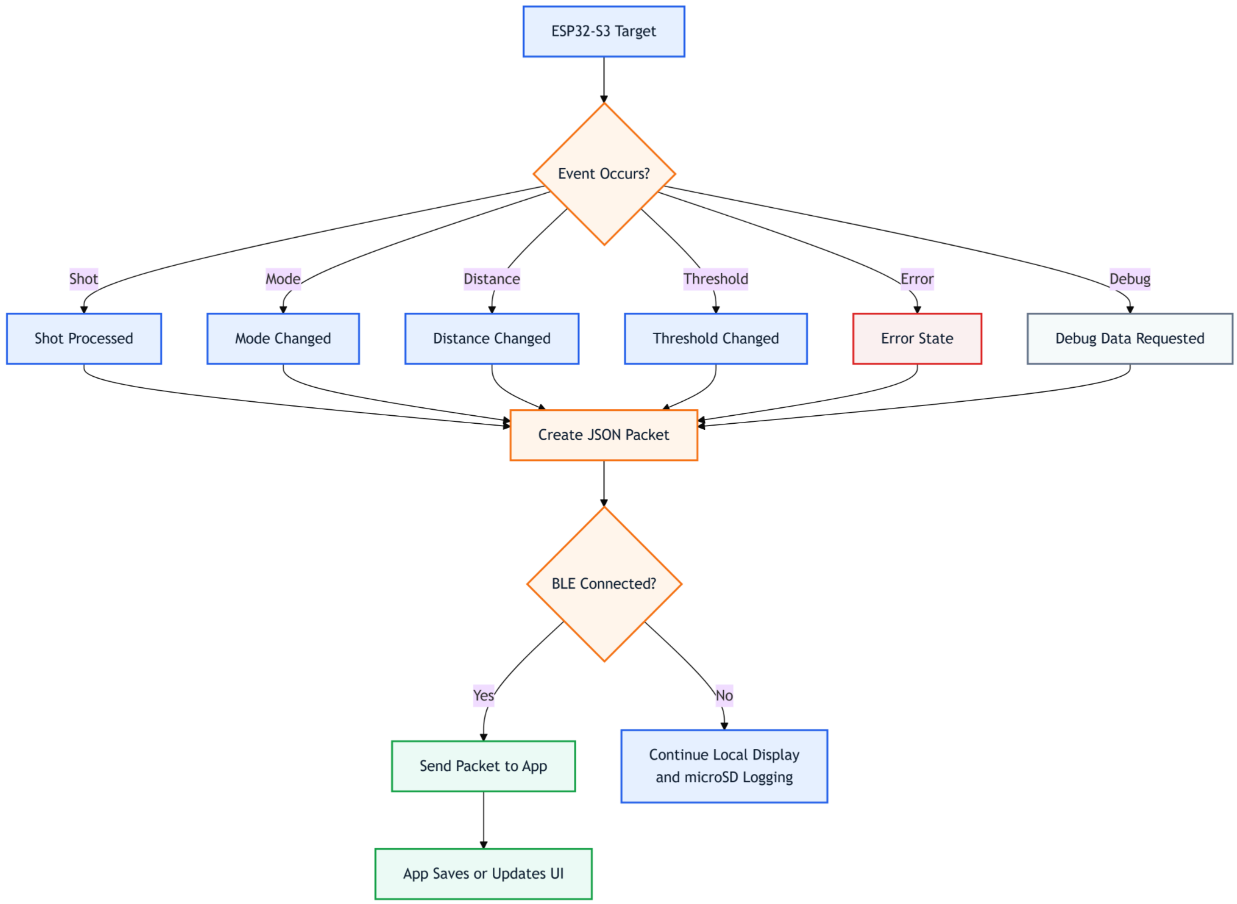


Figure 6.20: Wireless Data Flow

6.5 Reliability, Testing, and Future Expansion

6.5.1 Error Handling and System States

The software will need to handle normal operating states and error states clearly. The target should always know whether it is starting up, ready, training, connected to the app, calibrating, in debug mode, or in an error state. Keeping track of system states will make the firmware easier to organize and easier to debug.

Possible error conditions include multiple shots detected, invalid sensor readings, Bluetooth disconnection, microSD card failure, low battery, invalid mode selection, or calibration values that are too high or too low. When an error occurs, the target should show a simple message on the display and continue operating whenever possible.

For example, if the system detects two strong shot clusters at the same time, it can show a message such as “multiple shots detected” and reject that shot instead of calculating an incorrect score. If Bluetooth disconnects, the target should not stop working. It should continue detecting shots, updating the display, and logging locally if storage is available.

The goal of error handling is to prevent small issues from breaking the entire training system. The target should keep working as much as possible and clearly tell the user when something needs attention.

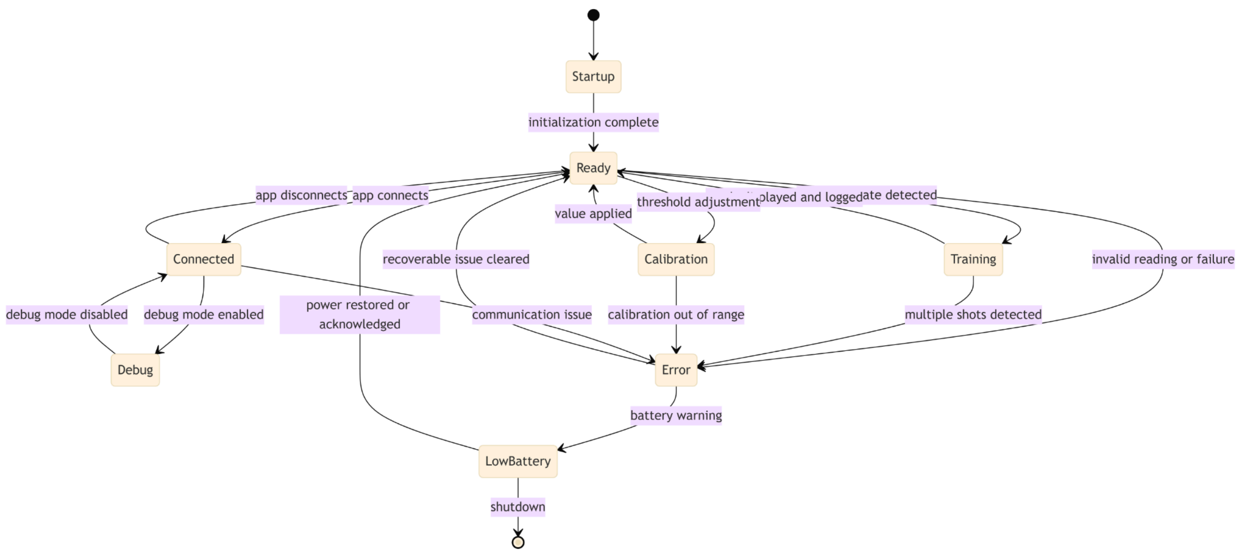


Figure 6.21: Error Handling and System States

6.5.2 Data Logging and Debugging System

The PrecisionShot system will include both normal logging and debug logging. Normal logging is used to save useful shot information, while debug logging is used to help the team test and improve the system during development. The mobile app will store user-facing shot history, while the microSD card can store device-side JSON logs.

For normal operation, each shot record may include the shot location, score, active mode, distance, timestamp, calibration value, and battery or connection state. This is enough information for the app to show history and analytics without storing too much extra data.

Debug mode will include more detailed information. This may include active sensor IDs, raw or filtered sensor values, threshold settings, calibration information, timing data, and firmware state. This information will help the team verify that the sensor array, multiplexer scanning, shot detection, scoring, and Bluetooth communication are working correctly.

The microSD logging system is planned to be always on during the prototype stage. If the logs become too large, the firmware may later include an automatic cleanup system that deletes older logs when storage becomes too full.

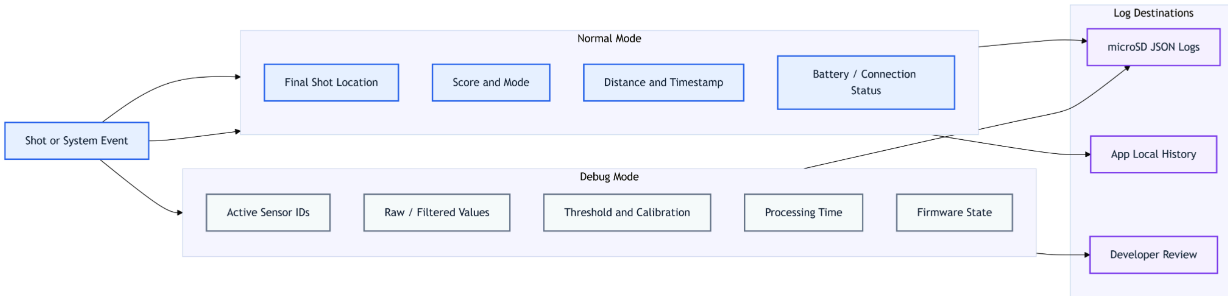


Figure 6.22: Data Logging and Debugging System

6.5.3 Testing and Debugging Logic

Testing will be performed throughout development to verify that each software feature operates correctly before the complete system is assembled. The major software functions will first be tested independently, including timer-based sensor scanning, shot detection, scoring, display updates, Bluetooth communication, mobile app updates, lighting calibration, microSD logging, and training-mode handling. After each subsystem works correctly on its own, the features will be tested together as one integrated system.

Shot-detection testing will be one of the most important parts of the software validation process. The team will compare known laser impact locations with the coordinates calculated by the ESP32-S3. This will verify the sensor mapping, strongest-response detection, nearby-sensor grouping, estimated shot-position calculation, and outlier rejection. Testing will also include weak signals, distant noise, neighboring sensor responses, repeated readings from one laser pulse, and multiple strong clusters within the same sensor frame.

Calibration testing will be performed under dark, normal, and bright indoor lighting conditions. The team will verify that the timer-based multiplexer system correctly records an individual ambient-light baseline for each of the 500 sensors. The global sensitivity setting will then be adjusted to confirm that the system rejects normal lighting changes while continuing to detect valid laser shots. Calibration controls on both the target and mobile application will also be tested to ensure that updated settings apply correctly.

Wireless testing will verify that shot results, mode changes, distance settings, calibration updates, and system-status information are transferred correctly between the target and mobile application. Bluetooth disconnection and reconnection will also be tested to confirm that the target continues operating independently and that the app retrieves the current system state after reconnecting.

Debugging will use serial output, mobile app diagnostic views, and microSD log files. Debug data may include sensor identification numbers, raw and filtered readings, ambient baselines, scan timing, detected clusters, rejected outliers, and current firmware state. These records will help the team identify whether a problem originates in the sensor array, multiplexer scanning, detection algorithm, scoring logic, display system, or Bluetooth communication.

Overall, testing will begin with individual software components and progress toward full hardware-and-software integration. The final testing process will verify that PrecisionShot can reliably detect and locate shots, calculate scores, update the onboard display, communicate with the mobile app, store records, and continue operating without a connected phone.

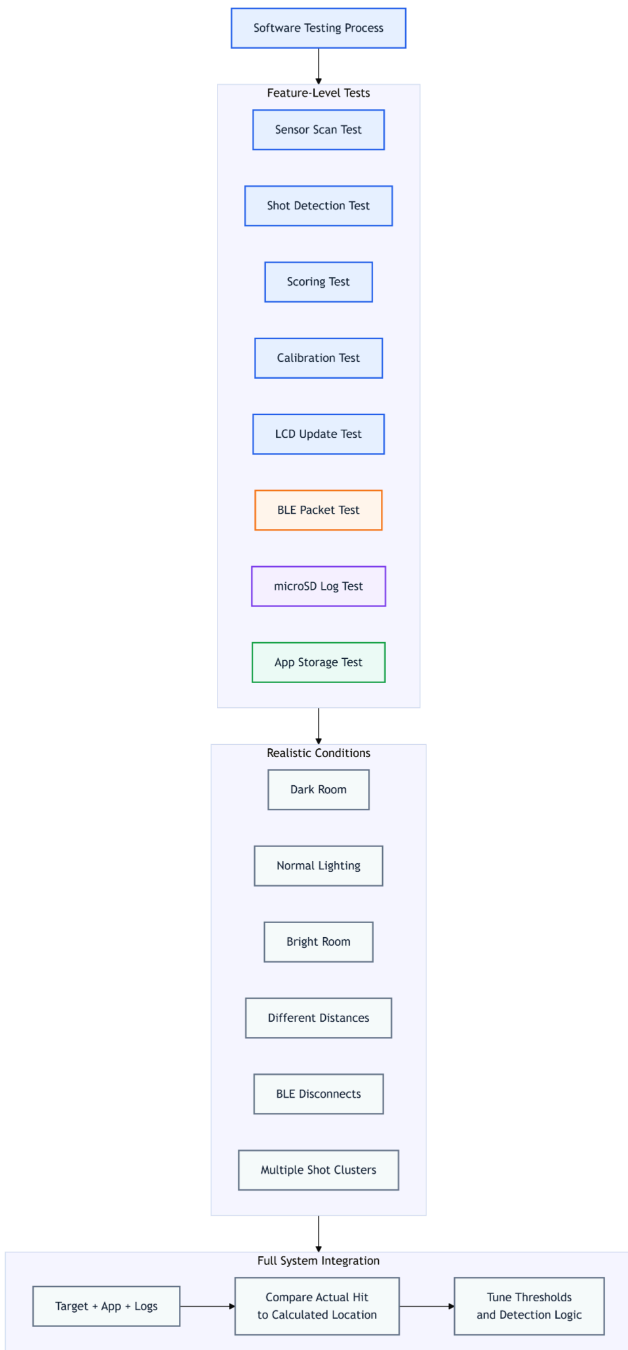


Figure 6.23: Testing and Debugging Logic

6.5.4 Future Software Expansion

Several software features may be added after the first prototype is completed. One possible expansion is support for multiple local user profiles. The prototype will use one local profile, but future versions could allow several users to maintain separate training histories, settings, and performance statistics on the same phone.

Another possible improvement is automatic distance measurement. For the first prototype, the user will enter the shooting distance manually through the target interface or mobile application. A future version could use a range sensor, camera, or another external device to measure the distance automatically. This feature is not included in the initial prototype because it would require additional hardware, calibration, and software development.

The Bluetooth communication protocol may also be improved. The prototype will use JSON packets because they are easy to inspect and debug. If testing shows that JSON packets are too large or slow, a later version may use a compact binary format to reduce transmission size and improve communication speed.

Future versions may also support continuously adaptive lighting calibration, advanced moving or randomized target modes, customizable training drills, and AI-assisted performance feedback. Additional app features could include exportable training records, optional user-controlled backups, and more detailed graphs and analytics.

These features are not required for the first working prototype, but the software architecture is designed so they can be added later without replacing the core shot-detection, scoring, storage, or communication systems.

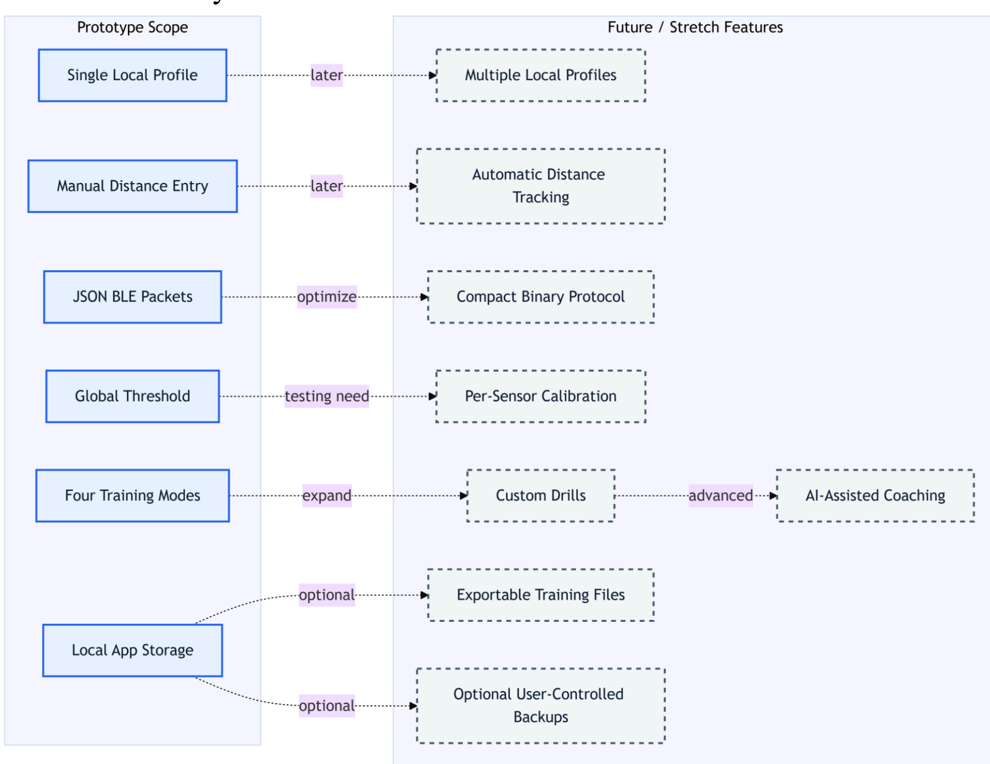


Figure 6.24: Future Software Expansion

Chapter 7: Prototype and Physical Design

7.1 Prototype Overview

When designing the physical layout of the PrecisionShot Training System, the primary focus centered on creating a rugged, portable enclosure that protects internal electronics while ensuring optimal performance from the light sensors. The physical design directly houses two of the three main system pillars: the phototransistor input array and the control module. Since this is an indoor training system meant to provide a cost-effective alternative to live-ammunition training, the housing must be practical, durable, and structurally optimized to ensure clean data delivery to the processing loop.

From an engineering perspective, balancing three main design priorities was critical to the success of the prototype:

Ambient Light Isolation: Regular indoor lighting can flood the sensors, making it difficult for the software's shot-detection logic to distinguish a true laser strike from background noise. To assist the calibration firmware, the housing must naturally shade the sensors from the surrounding environment. The design addresses this by incorporating a front-facing grid mesh directly over the sensors. This mesh structure restricts light entry exclusively to direct, front-angled paths while completely blocking ambient light bleed from the sides. The visible target pattern is printed directly onto the face of this mesh for the user to aim at.

Structural Rigidity: The chassis must possess enough structural stiffness to completely resist twisting, flexing, or warping. Any physical flexing within the frame shifts the coordinate alignment of the sensor grid, which would directly compromise the spatial positioning accuracy and score calculation logic within the software.

Easy Access to Components: The enclosure architecture incorporates a modular, serviceable rear panel rather than a permanently sealed unit. This design provides direct access to internal subsystems for hardware-in-the-loop debugging, test-point voltage validation, firmware flashing, and routine battery maintenance without requiring a destructive or complex disassembly of the target faceplate grid.

7.2 Target Layout

To optimize the interaction between the physical user interface and the optical detection hardware, the internal area of the target chassis is divided into two primary functional zones. The overall physical frame supports a main PCB footprint measuring 20 inches vertically by 14 inches horizontally. The lower section of this footprint allocates a 14-inch-tall by 14-inch-wide region specifically for the active target capture zone. The remaining upper 6 inches of the chassis are dedicated to the control module, accommodating the user-interface buttons, routing paths, power-management components, and digital LCD panel. This vertical separation ensures that the display components and control inputs remain visible and separated from the active target area during training sessions.

The active tracking region within the lower section of the chassis features a full 20-column by 25-row rectangular matrix measuring 12.5 inches in height by 10.0 inches in width and operating on a uniform 0.5-inch grid pitch. The 500 active phototransistor sensors are each co-located with a dedicated blue surface-mount visual-feedback LED, providing one-to-one optical sensing and illumination at every coordinate node. The 500 active tracking pairs are divided among five concentric scoring zones that define the target hierarchy. The physical sensor spacing remains uniform across all five zones, while the zone boundaries determine the score assigned to each detected coordinate.

The 500 active tracking nodes are distributed across the five concentric scoring zones according to the following breakdown:

- Zone 1 (Center Core / Red): Positioned at the geometric center of the active tracking region, this inner rectangular scoring area contains 120 sensors, representing 24.0% of the total array.
- Zone 2 (Inner Ring / Green): Located immediately outside the central red core, this first concentric scoring region contains 112 sensors, representing 22.4% of the total array.
- Zone 3 (Middle Ring / Purple): Situated within the middle scoring band of the target, this region contains 104 sensors, representing 20.8% of the total array.
- Zone 4 (Outer Ring / Pink): Serving as the secondary outer scoring region, this zone contains 88 sensors, representing 17.6% of the total array.
- Zone 5 (Border Rim / Blue): Defining the outermost boundary of the 12.5-inch by 10.0-inch target matrix, this region contains 76 sensors, representing 15.2% of the total array. It allows the system to detect edge impacts and off-center shots.

By mapping each sensor to a known coordinate pair and scoring zone, the firmware can translate measured sensor signals into shot coordinates and zone-specific scores for display on the user interface.

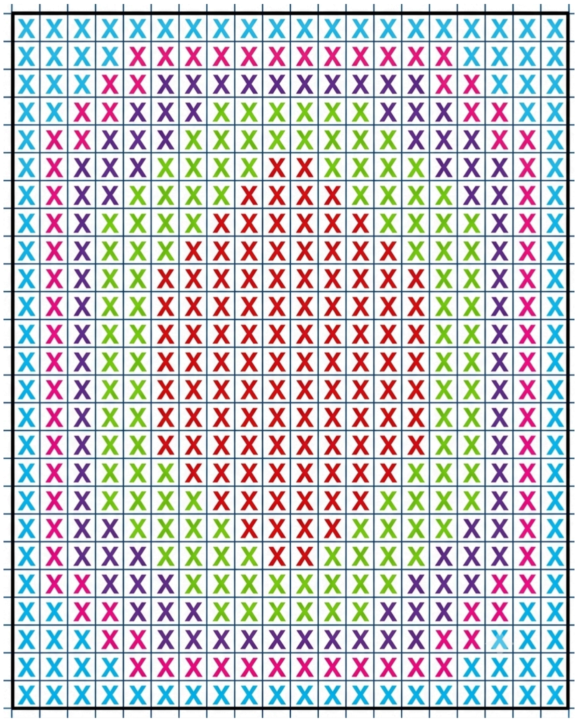


Figure 7.1: Target Sensor Zone Layout and Sensor Count Distribution

Table 7.2: Target Sensor and LED Distribution by Zone

Zone #	Color	Target Area / Rings	Sensor/LED Count	Percentage of Array
Zone 1	Red	Center Target Core	120	24.00%
Zone 2	Green	Inner Ring	112	22.40%
Zone 3	Purple	Middle Ring	104	20.80%
Zone 4	Pink / Magenta	Outer Ring	88	17.60%
Zone 5	Light Blue	Border / Perimeter	76	15.20%
TOTAL	—	Full Target Matrix	500	100.00%

7.3 Physical Enclosure Design

The physical enclosure must protect the large 20 in × 14 in internal PCB while remaining easy to mount and transport. The front faceplate acts as an optical filter for incoming laser light, while the outer enclosure matches the 20 in × 14 in dimensions of the internal PCB.

- Alignment Windows: Each aperture has an internal diameter of 2.5 mm and is centered over the 1.6 mm × 0.8 mm footprint of the underlying ALS-PT19-315C/L177/TR8 surface-mount phototransistor.
- Optical Isolation: The faceplate has a uniform thickness of 6.1 mm. Recessing the surface-mounted phototransistors behind these mechanical tunnels creates an angular shroud around each sensor node. This geometry limits the optical reception cone and helps block off-axis ambient room light from reaching the sensors.

- Rear Vents: The rear enclosure includes 26 ventilation slits, each approximately 6.35 mm wide, to promote airflow through the device and reduce the risk of overheating.

Because of the enclosure's large physical dimensions, the rear housing will be fabricated as a modular, 3D-printed structural frame. Internal reinforcement ribs run vertically across the back plate to hold the custom PCB securely against the faceplate and maintain planarity across the full 20-inch span. The enclosure will also provide access to the internal components and include a removable panel for servicing or replacing the battery when necessary. If time allows, an external charging port will be added so the device can be recharged without opening the enclosure.

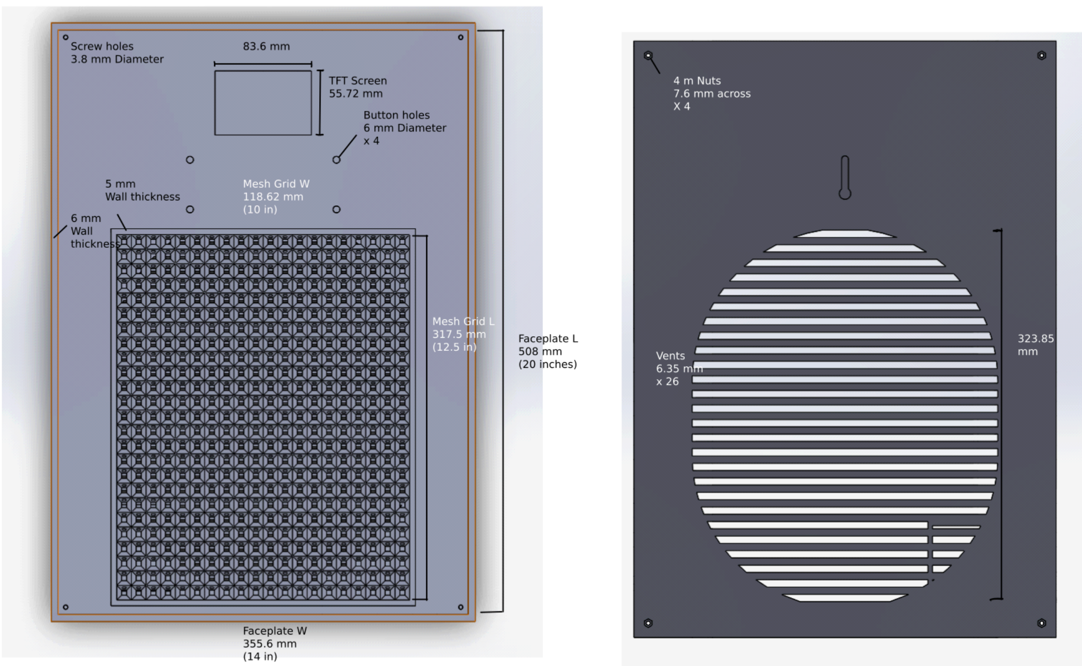


Figure 7.3: CAD Design - Target Layout and Component Placement

To satisfy the optical-isolation and structural-durability requirements of the target assembly, the physical enclosure integrates a custom front-facing grid plate. Instead of using a standard flat-panel layout, the target face contains a dense geometric matrix arranged in a 20-column by 25-row configuration, providing 500 active sensor positions across the light-capture area. Each cell within the matrix is a uniform square measuring 12.7 mm by 12.7 mm.

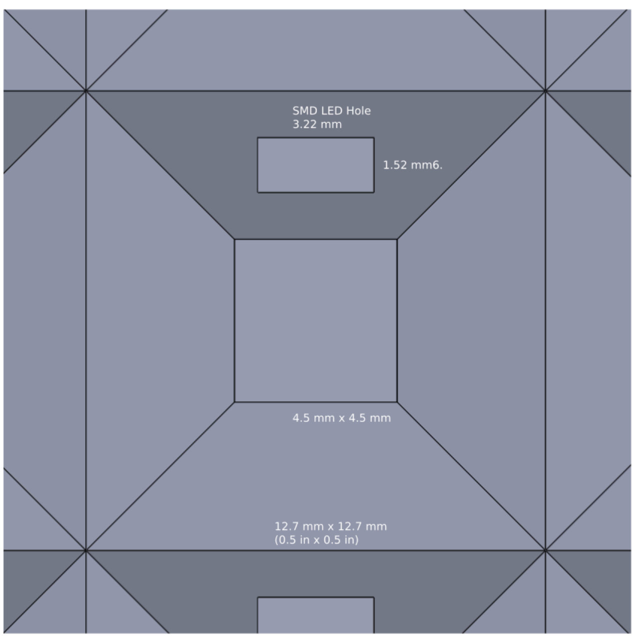
Rather than using straight vertical dividing walls, all four inner sides of each cell slope inward at a 45-degree angle and extend 4.1 mm into the 6.1 mm-thick faceplate. This geometry forms a series of inverted square pyramids, or chamfered funnels, while the remaining 2 mm forms the base of each cell. The angled walls help guide incoming laser light toward the sensor aperture while limiting the amount of off-axis ambient light that can reach the phototransistor. The 45-degree geometry also allows the grid plate to be 3D printed without requiring internal support structures.

At the bottom and geometric center of each funnel is a flat 4.5 mm by 4.5 mm aperture shelf. A smaller opening centered within this shelf aligns with the underlying printed circuit board, allowing the optical surface of the corresponding phototransistor to receive incoming light. To provide immediate localized feedback after a successful hit without introducing significant optical interference, an indicator light-emitting diode is integrated into the upper sloped wall of each chamfered cavity.

Rather than remaining exposed on the target surface, the LED is recessed within a dedicated 3.2 mm-diameter tunnel designed for a standard 3 mm LED package. This tunnel passes through the 45-degree sloped wall and directs the feedback illumination forward toward the user while reducing lateral and downward light leakage toward the phototransistor located at the base of the cell.

Mechanically, the grid plate maintains a uniform total thickness of 6.1 mm along the Z-axis. This includes the 4.1 mm chamfered collection region and the remaining 2 mm structural base. The resulting thickness provides rigidity across the large multi-sensor panel and helps reduce warping during fabrication and assembly.

To improve optical isolation across all 500 sensor nodes, the grid plate will be fabricated from an opaque, dark-colored material. A minimum solid-wall clearance of approximately 1.3 mm is maintained between the LED tunnel and the outer boundary of the chamfer. This material separation helps reduce internal light bleed and prevents the feedback LEDs from falsely triggering nearby phototransistors during system operation.



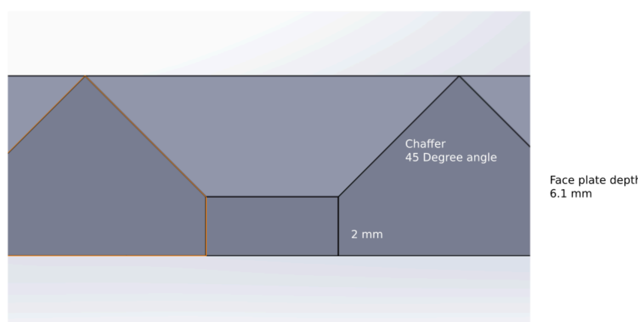


Figure 7.4: Individual Cell Design For The Grid

7.3.1 Optical Diffuser Material Selection and Decision Analysis

While the 45-degree chamfered funnel grid provides mechanical shielding and directional laser channeling, placing a bare, uncovered sensor array directly exposed to the environment presented two primary challenges: localized point-source glare from the blue feedback LEDs (hot-spotting) and surface reflections off the phototransistor lenses. To achieve a smooth visual feedback surface while maintaining maximum laser detection sensitivity, an optical diffuser layer was evaluated and integrated directly over the 12.5 in x 10.0 in active target face.

During the material evaluation phase, two primary acrylic diffuser types were evaluated against system performance parameters:

Frosted / Matte Clear Cast Acrylic (P95 Finish): Evaluated for its high forward optical transmission and non-glare surface properties. While effective at suppressing ambient surface reflections, its high transparency allowed distinct LED hot-spots to remain visible through the faceplate, preventing clean zone-wide illumination.

Translucent White / Opal Cast Acrylic Sheet (Style 2447, 55% Transmittance): Evaluated for its superior backlight diffusion properties. This sign-grade 1/8-inch (3.0 mm) thick material completely blends individual LED emission cones into uniform, softly illuminated target blocks without individual diode hot-spots.

Experimental testing confirmed that the 1/8-inch Translucent White 55% transmittance acrylic provided the optimal balance for the PrecisionShot architecture. The 55% light transmission factor allows sufficient optical energy from incoming laser strikes to pass through the sheet and hit the phototransistor array above threshold levels, while providing uniform visual diffusion across all five target zones when the feedback LEDs illuminate.

Mechanically, a 12.0 in x 24.0 in x 1/8 in stock panel is cut to fit precisely over the active matrix, serving a dual purpose as an optical homogenizer and a protective front shield that seals the internal phototransistor cavities against dust and environmental contamination.

7.4 Control Subsystem and User Interface Integration

The upper 6.0-inch panel of the enclosure serves as the command console for the system and houses the main user controls and visual display. Positioning the control subsystem above the target array separates the display, buttons, power wiring, and other switching signals from the high-density phototransistor and multiplexer circuitry below. This arrangement reduces wiring congestion, improves accessibility, and helps prevent electrical noise from interfering with the low-level sensor signals.

The user interface is centered around an integrated 4.0-inch, 320 × 480 TFT LCD module with capacitive touch capability. The display presents the active training mode, shot count, most recent score, Bluetooth connection status, battery information, calibration status, and other system settings. The touchscreen provides the primary method for navigating menus and controlling software features, while the physical buttons provide direct access to important functions.

Mechanical Fit: The front faceplate contains a rectangular cutout measuring 120.0 mm × 76.0 mm with an internal bezel lip. The display module is installed from the rear so its viewing area sits flush against the bezel. Integrated mounting bosses secure the module using four M2.5 non-conductive nylon screws. This mounting method keeps the screen aligned, prevents movement during operation, and allows the module to be removed for repair or replacement.

Electrical Routing: The TFT module uses an onboard ST7796S display controller, allowing the ESP32-S3 to communicate with the screen through a high-speed SPI interface rather than a large parallel data bus. The connection includes SPI clock and data signals along with dedicated chip-select, data/command, reset, and backlight-control lines. The capacitive touch controller communicates separately through the I2C bus and uses touch-interrupt and reset signals where required. This arrangement reduces the number of ESP32-S3 GPIO pins needed and allows the display wiring to route cleanly without interfering with the sensor matrix and multiplexer connections.

Directly adjacent to the LCD screen, three tactile interface buttons are arranged horizontally for fast access to essential system functions. Each button is positioned within a properly toleranced circular guide opening to prevent binding, misalignment, or accidental activation. The buttons connect directly to ESP32-S3 GPIO inputs and use firmware debouncing to prevent a single press from being registered multiple times.

Mode Selection Button: This button allows the user to cycle through the available firmware training modes, such as Classic, Freestyle, and Rapid. When pressed, the firmware updates the active operating mode, resets any mode-specific variables as needed, and refreshes the TFT display to show the new selection.

Session Reset Button: This button clears the active session variables, shot counters, score data, and temporary result arrays stored in the ESP32-S3's working memory. It allows the user to begin a new session without restarting the system or interrupting the Bluetooth connection with the companion mobile application.

Ambient Calibration Button: This button initiates a complete baseline scan across all 500 phototransistor sensors. During calibration, the ESP32-S3 records the ambient-light level of each sensor and updates the detection thresholds used by the shot-detection firmware. This allows the system to compensate for changes in indoor lighting and maintain reliable laser detection without requiring a full power cycle or manual adjustment.

7.5 Mounting and Assembly

Flawless alignment between the 500 phototransistors and the faceplate apertures is achieved by surface-mounting the sensors directly to the top layer of our custom board. The board is then set into place aligned with the faceplate using a 40 mm screw and a 4 M nut that will fit perfectly into the back of the enclosure.

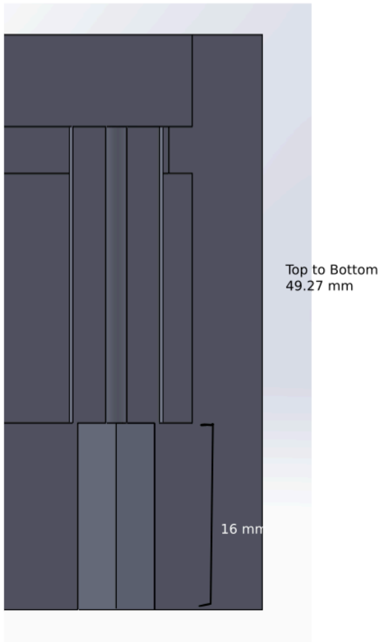


Figure 7.5: Fastening cross section

To accommodate different training environments, the target chassis features integrated mechanical mounting points for both wall-mounted and tabletop deployment. The rear panel of the enclosure includes a set of recessed, reinforced keyhole slots spaced to fit standard mounting brackets or wall-anchor hardware. This design allows the entire target unit to hang securely on a vertical surface while keeping the assembly flush against the wall to minimize movement when struck. For flat-surface deployments, such as placing the target on a table or workbench, the bottom face of the chassis is designed with a wide, stable footprint equipped with integrated slots for non-slip rubber feet. This prevents the unit from shifting, tilting, or sliding on smooth surfaces during operation.

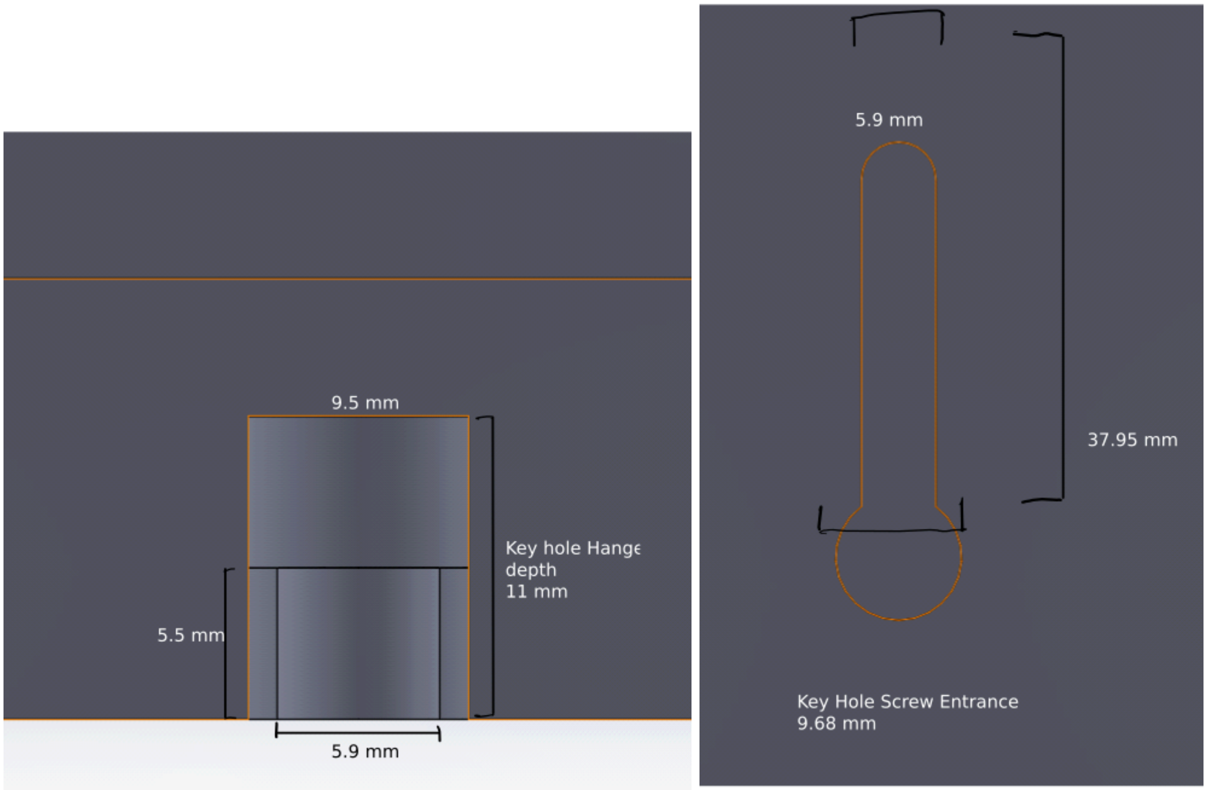


Figure 7.6: Key hole hanger cross section and dimensions

Because the enclosure houses a large custom PCB alongside a heavy semi-solid-state power bank, the interior of the chassis utilizes an array of molded structural supports to prevent internal component flexing. The main sensor array PCB sits on top of a perimeter shelf equipped with internal standoffs that elevate the board away from the outer walls, protecting the trace layers and soldered joints from compression. The internal containment bracket for the power bank will be reinforced with vertical ribbing to handle the weight of the cells without warping over time. The entire assembly is fastened together using a series of strategically placed, threaded screw.

The power bank will sit comfortably behind the PCB in a slot designed to be snug with a slot that will allow the built in power cable from the bank to feed directly into the microcontroller. A small slot will be left so that the power bank can be charged while still in the enclosure.

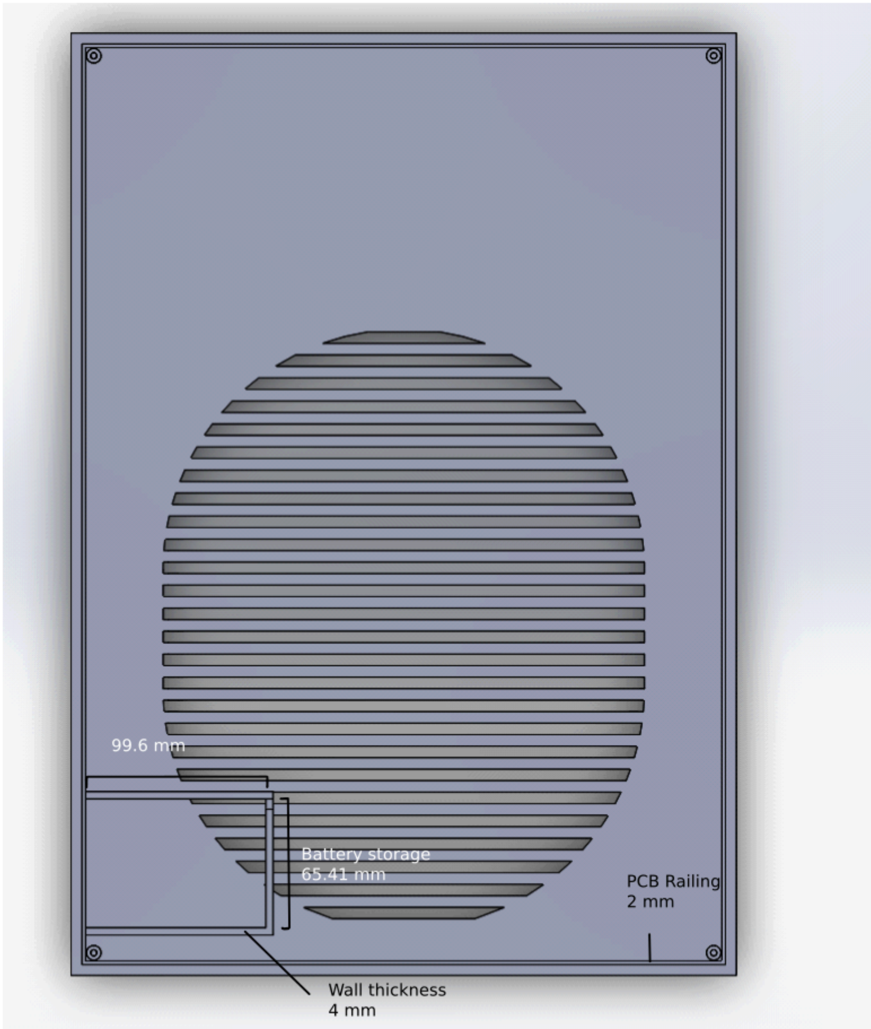


Figure 7.7: Battery storage compartment in the back of the enclosure

Maintaining absolute mechanical alignment between the front-facing grid plate and the underlying sensor PCB is critical to prevent optical misalignment. To achieve this, the front bezel incorporates internal alignment pins that mate directly with matching alignment holes on the PCB. This mechanical constraint ensures that every 10.5 mm by 10.5 mm chamfered cell remains dead-centered over its corresponding 4.5 mm by 4.5 mm sensor aperture shelf during the final tightening phase. For routine maintenance or component troubleshooting, the enclosure is designed for rear-panel access. The back faceplate functions as a removable hatch, secured by recessed hex fasteners. Removing this panel exposes the back of the main PCB, the power bank bracket, and the input routing cables. This structural layout ensures that the front grid faceplate and optical paths remain completely sealed and undisturbed if internal electronics ever need to be inspected, repaired, or replaced.

The upper 6-inch segment of the enclosure frame contains our primary user control interfaces, cleanly separating them from the lower stadium scoring zone:

- Power Switch: To prevent parasitic drain when the device is stored, a physical Single-Pole Single-Throw (SPST) rocker switch is integrated into the primary power rail. Because the selected KU XIU Semi-Solid-State battery subsystem outputs its stable power envelope via an integrated USB-C distribution hub or discrete standard output pins, a traditional inline battery-cutoff switch cannot be spliced directly into the raw, internal pouch terminals.

  The power switch is implemented directly on the system's main PCB, inline with the positive voltage trace immediately following the incoming power connector. When the rocker switch is toggled to the *off* position, it physically breaks the incoming 5 Volt rail.

- USB-C Charging Port:To facilitate safe, high-speed recharging without increasing PCB complexity, the custom housing includes a precise mechanical opening that exposes the primary USB-C input interface on the KU XIU Semi-Solid-State battery.

  This design choice leverages the factory-calibrated over-current and thermal management protection loops embedded directly inside the gel-polymer battery assembly. By securing the battery pack structurally against the side panel of the 3D-printed enclosure, a standard external USB-C cable can pass seamlessly through the chassis cutout to mate directly with the internal controller. This configuration yields a highly durable, foolproof charging link that safely restores the device's internal power reservoir at factory-rated speeds without introducing unnecessary electronic failure points or thermal sinks onto the main target sensor PCB.

- Interface Navigation Buttons: Located directly below the screen, these buttons send physical hardware interrupts to the control module, letting the user toggle between training modes, reset scores, and trigger manual ambient-light calibration routines.
- LCD Display Window: A clean, rectangular cutout houses the primary digital display module. This screen outputs real-time feedback from the processing loop, showing shot coordinates , active training modes , scoring data , and real-time telemetry from the battery power hub. To achieve high accuracy and stable capacity readings, the power monitoring subsystem utilizes a Coulomb Counting algorithm implemented via firmware tracking over the INA219 high-side monitor. The graphical user interface layer is optimized to display the advanced performance analytics calculated by the embedded processing loop. When a hit is processed, the display maps and renders a localized visual representation of the target face, plotting the precise position of the laser impact alongside a running tally of shot counts and session total scores. Depending on the configured training profile, the UI seamlessly transitions its format: presenting simple coordinates in Freestyle mode , center-distance matrices in Classic mode , or dynamic countdown bars and target-acquisition response timers during fast-paced Rapid or Moving Target drills. This deep hardware-to-software convergence provides the user with an intuitive, standalone dashboard that delivers robust, live multi-metric training statistics completely independent of the mobile companion application

7.6 Prototype Illustrations

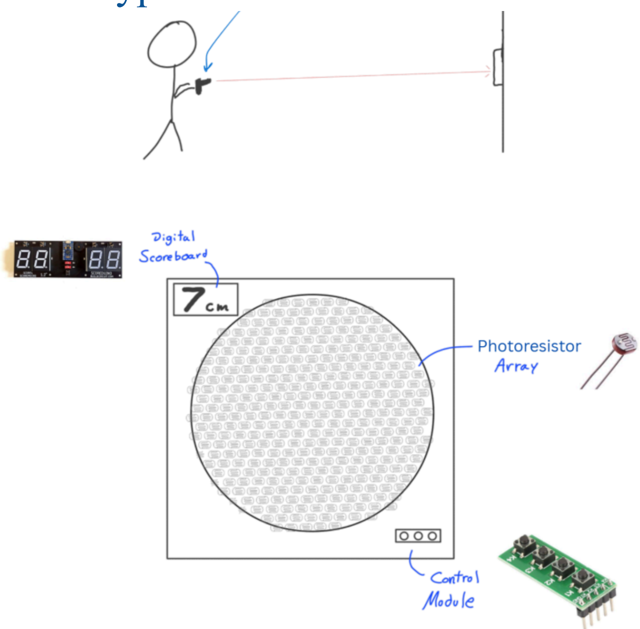


Figure 7.8: Initial Concept Drawing

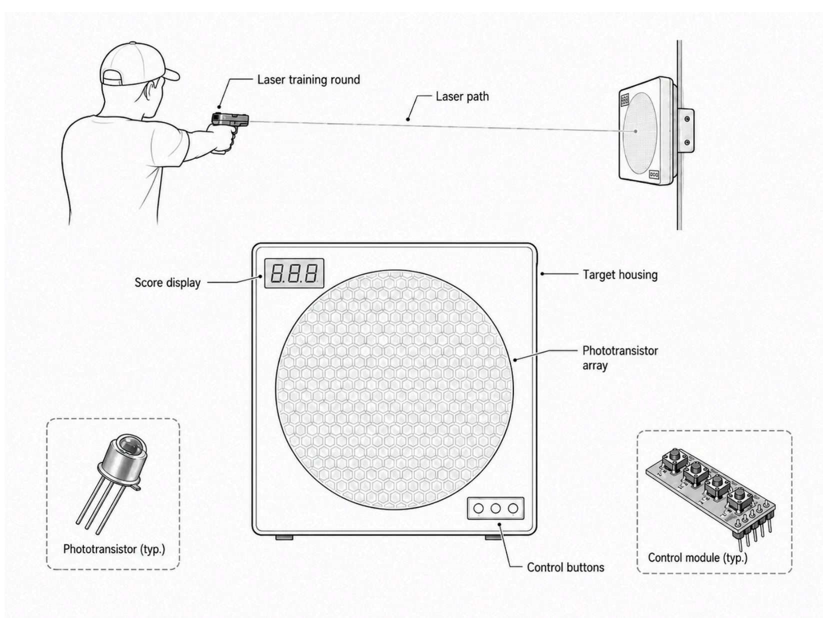


Figure 7.9: AI-Assisted PrecisionShot System Concept and Component Layout

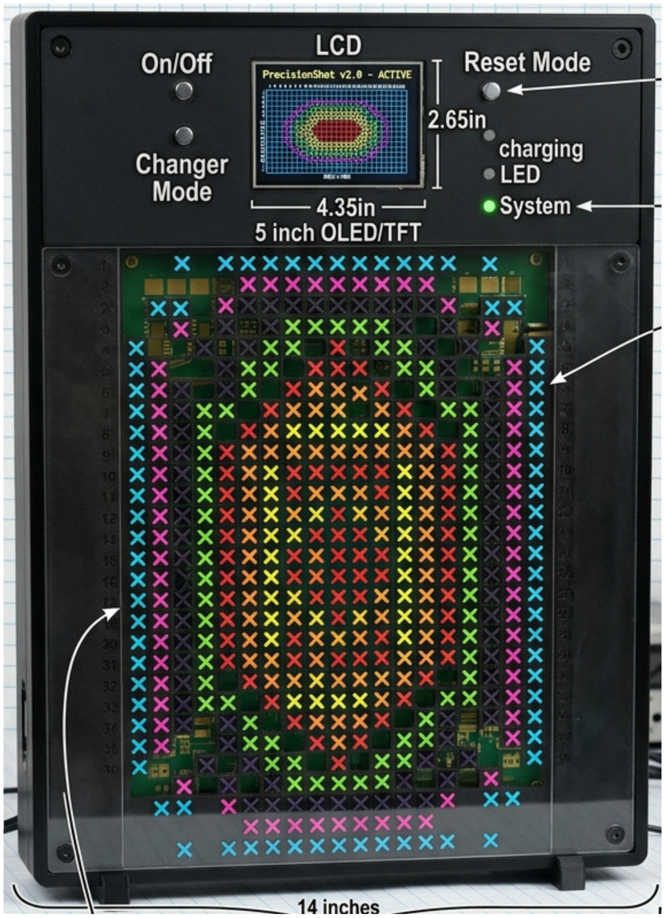


Figure 7.9: AI-Assisted Concept Rendering of the *PrecisionShot* Training System

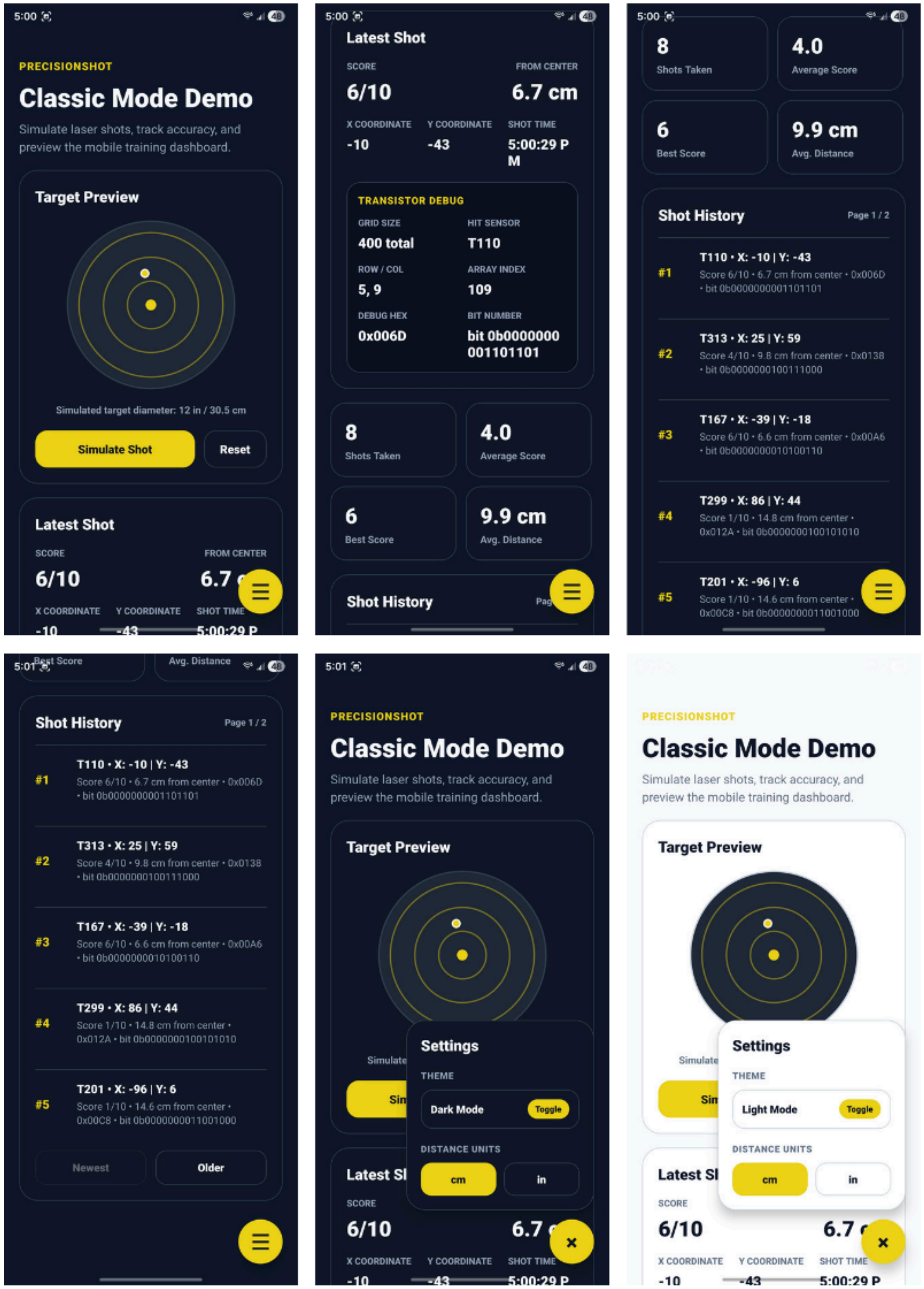


Figure 7.10: *PrecisionShot* Mobile Application User Interface Demonstration

Chapter 8: Budget and Financing

8.1 Budget Overview

The current estimated cost of the first working PrecisionShot prototype is approximately $922.20. This estimate includes the sensor and LED feedback arrays, multiplexing and control electronics, custom PCB fabrication, onboard display, power system, local storage, user controls, enclosure materials, laser-training cartridge, wiring, connectors, shipping, taxes, and a contingency allowance for prototype revisions.

The largest expected expenses are the custom PCB and stencil, phototransistor array, enclosure and faceplate materials, and prototype contingency allowance. These costs are significant because PrecisionShot requires a large number of sensing and feedback components, a high-density custom circuit board, and a mechanically aligned enclosure that positions each sensor correctly beneath the target grid. Additional costs include the ESP32-S3 controller, analog multiplexers, LED drivers, display, microSD storage, power regulation, battery monitoring, switches, and internal wiring.

A contingency allowance of $100 is included to account for shipping, taxes, replacement components, assembly mistakes, enclosure reprints, and minor hardware revisions. Prototype development commonly requires adjustments after initial testing, particularly when electrical performance and mechanical alignment must be verified together.

Overall, the estimated cost is reasonable for a custom Senior Design prototype that combines dense optical sensing, embedded processing, localized LED feedback, wireless communication, onboard controls, rechargeable power, and a purpose-built enclosure. The budget will continue to be updated as supplier prices are confirmed, components are purchased, and the final design is prepared for fabrication.

8.2 Component Cost Breakdown

Table 8.1: Prototype Component Cost Breakdown (BOM)

Component	Quantity	Unit Cost	Total
ALS-PT19-315C/L177/TR8 phototransistors	500	$0.35	$175.00
IN-S126ATB surface-mount feedback LEDs	500	$0.05	$25.00
74HC4067 16-channel analog multiplexers	34	$1.25	$42.50
Phototransistor bias resistors	500	$0.01	$5.00
LED current-limiting resistors	500	$0.01	$5.00
16-channel LED driver ICs	32	$1.75	$56.00
Multiplexer and driver decoupling capacitors	70	$0.05	$3.50
Sensor filtering and signal-conditioning components	1 lot	$15.00	$15.00

ESP32-S3-WROOM-1-N8R8 control module	1	$15.00	$15.00
Hosyond 4.0-inch ST7796S touchscreen module	1	$25.00	$25.00
microSD card socket or interface module	1	$8.00	$8.00
microSD storage card	1	$7.00	$7.00
TLV62569DBVT 3.3 V buck regulator	1	$2.50	$2.50
Buck-regulator power inductor	1	$2.00	$2.00
Regulator capacitors and feedback resistors	1 set	$5.00	$5.00
INA219 battery and current monitor	1	$10.00	$10.00
Resettable input polyfuse	1	$1.00	$1.00
USB-C power-input receptacle	1	$2.00	$2.00
KUXIU 5000 mAh USB-C power bank	1	$30.00	$30.00
USB-C power cable	1	$8.00	$8.00
NKK EB2011-BG panel-mounted control switches	4	$3.80	$15.20
Rotary encoder and adjustment knob	1	$4.50	$4.50
Headers, JST connectors, wiring, and internal cables	1 lot	$25.00	$25.00
Main sensor and control PCB	1	$150.00	$150.00
PCB solder-paste stencil	1	$25.00	$25.00
Solder paste and assembly materials	1 lot	$15.00	$15.00
3D-printed enclosure and faceplate material	1 set	$60.00	$60.00
Acrylic, diffuser, and grid-filtering material	1 set	$25.00	$25.00
Screws, standoffs, threaded inserts, and mounting hardware	1 set	$20.00	$20.00
Mantis Pink Rhino laser-training cartridge	1	$40.00	$40.00
Shipping, taxes, and prototype contingency	1 lot	$100.00	$100.00
Prototype Total			$922.20

The component cost breakdown lists the electrical, mechanical, optical, and assembly materials required to build one complete PrecisionShot prototype. The largest cost contributors are the custom PCB, phototransistor array, enclosure, LED feedback system, and shipping and prototype contingency.

The total prototype cost is $922.20. This amount includes the complete sensor and LED arrays, processing and communication hardware, display, local storage, battery system, laser cartridge, enclosure, connectors, assembly materials, and mounting hardware.

8.3 Cost Assumptions and Budget Risk

The prototype budget is based on the construction of one complete working PrecisionShot system rather than multiple finished units. The estimate includes the major electronic, optical, mechanical, and power-system components required for the current 500-sensor design. Common laboratory equipment, soldering tools, test instruments, software, and engineering labor are not included because they are expected to be available through the university or team members as part of the Senior Design development process.

The estimate assumes that the major design choices will remain generally consistent through final assembly. These include the 500-phototransistor and 500-LED arrays, ESP32-S3 controller, multiplexer network, display, power bank, voltage-regulation system, microSD storage, and custom enclosure. Supplier prices, shipping charges, taxes, and component availability may still change before all purchases are completed.

The PCB and solder-paste stencil estimates are based on prototype fabrication and stencil-ordering services available from PCBWay [23], [24]. The most significant financial risks are additional fabrication runs, enclosure revisions, damaged components, and unexpected hardware substitutions. Errors in the schematic, PCB routing, component footprints, or sensor mapping could require another board order. Mechanical alignment problems between the sensors, LEDs, faceplate, and enclosure could also require additional printed parts or material. Component shortages or longer-than-expected shipping times may force the team to purchase more expensive alternatives.

To reduce these risks, the team will validate major subsystems before completing the final assembly. Breadboard testing, small-scale sensor and multiplexer testing, power-system verification, display testing, and early enclosure fit checks will be used to identify problems before committing to the full prototype. The budget includes a $100 contingency allowance for shipping, taxes, replacement parts, assembly mistakes, and minor prototype revisions.

8.4 Funding and Sponsorship

The PrecisionShot Training System is currently planned as a self-funded project. Based on the estimated prototype cost of $922.20, an equal division among the four team members would result in an estimated contribution of approximately $230.55 per member. This makes the project financially achievable without depending on outside sponsorship.

The team may also use university resources such as laboratory equipment, test instruments, technical guidance, fabrication facilities, and assembly tools. These resources can reduce the need to purchase specialized equipment or outsource development work. Any donated materials, department support, or other outside assistance would reduce the team's personal expenses, but no external funding is included in the current budget.

The primary financial objective is to complete and validate one functional prototype. Available funds will therefore be prioritized toward the custom PCB, sensor and LED arrays, enclosure construction, power-system stability, control electronics, and final system integration. Cosmetic improvements, optional accessories, and advanced features will only be considered after the core engineering requirements have been satisfied.

8.5 Preliminary Mass Production Cost and Margins

The current budget represents the cost of building a single engineering prototype and should not be treated as a production-unit cost. Prototype expenses include small-quantity component pricing, individual shipping charges, replacement parts, design revisions, and fabrication methods selected for flexibility rather than high-volume efficiency.

If PrecisionShot were manufactured in larger quantities, some costs could decrease through bulk component purchasing, panelized PCB fabrication, automated component placement, standardized assembly procedures, and reduced per-unit shipping costs. However, a defensible manufacturing cost cannot be calculated until the final schematic, PCB layout, enclosure design, bill of materials, assembly process, and testing requirements have been completed and validated.

The enclosure would require particular manufacturing analysis because the current design relies on three-dimensional printing. This method is appropriate for prototype development because it supports rapid revisions without requiring permanent tooling. Larger production volumes may justify alternative methods that improve consistency and reduce assembly time, but those methods would require additional tooling costs and design-for-manufacturing changes.

A commercial version of PrecisionShot would also include costs not represented by the prototype bill of materials. These may include assembly labor, quality-control testing, packaging, documentation, regulatory and laser-safety review, software maintenance, warranty service, fulfillment, inventory storage, returned units, customer support, and product marketing.

Because manufacturer quotations, production tooling costs, labor rates, compliance requirements, and market demand have not yet been established, the team is not assigning a specific mass-production cost, retail price, or expected profit margin. A future commercialization study would require a finalized bill of materials, supplier and manufacturer quotes, a design-for-manufacturing review, safety and compliance evaluation, competitor analysis, and a detailed business plan.

The current priority remains the successful construction and validation of the Senior Design prototype. Commercialization may be considered after the system demonstrates acceptable accuracy, reliability, usability, and manufacturing feasibility.

Chapter 9: Project Milestones and Timeline

9.1 Timeline Priority Definitions

To organize development and reduce the risk of schedule delays, each PrecisionShot milestone is assigned a priority level based on its importance to the operation, validation, and final demonstration of the system. Priority levels are determined by technical dependency, project risk, and whether the milestone is required to satisfy the core engineering specifications.

High-priority milestones are required for the prototype to function and must be completed before optional features are pursued. These milestones include phototransistor and multiplexer testing, PCB design and assembly, ESP32-S3 firmware development, power-system validation, shot-detection and scoring logic, onboard feedback, Bluetooth communication, enclosure alignment, system integration, and final performance testing. Delays in these areas could prevent the team from completing or demonstrating a functional prototype.

Medium-priority milestones improve the usability, reliability, testing capability, or presentation quality of the system but are not required for its most basic operation. These milestones include mobile application analytics, local shot-history storage, microSD logging, expanded debugging tools, interface refinement, enclosure improvements, and further development of the planned training modes. Medium-priority work will begin after the related high-priority subsystem is operating reliably.

Low-priority milestones are stretch features that may improve the long-term user experience but are not required to meet the primary Senior Design objectives. These include automatic distance measurement, customizable drills, advanced coaching feedback, shot-angle tracking, optional cloud backup, and additional application visualizations. Low-priority features will only be attempted if the core prototype has been completed, tested, and shown to operate consistently.

Milestone priorities may be adjusted during development if testing reveals new technical risks or dependencies. When schedule or budget limitations occur, the team will prioritize work that directly supports reliable shot detection, accurate feedback, standalone operation, and successful final demonstration.

9.2 Senior Design 1 Milestones

Table 9.1: Senior Design 1 Milestones and Completion Criteria

Week / Period	Priority	Milestone	Description / Completion Criteria
Week 1	High	Form project group	Confirm team members, roles, communication methods, and project expectations.
Week 1	High	Select project concept	Choose PrecisionShot as the final project idea and define it as a smart laser training system rather than only a basic target.
Week 1–2	High	Define project motivation and goals	Establish the main purpose of the system, including safe dry-fire training, shot detection, real-time feedback, portability, and app integration.
Week 2	High	Complete early product research	Research existing laser training products and identify limitations that PrecisionShot should improve.
Week 2	High	Create initial goals and objectives	Separate project goals from measurable engineering objectives.
Week 2–3	High	Complete Divide and Conquer report	Submit the initial project planning document and prepare for the first group meeting.
Week 3	High	Attend D&C group meeting	Receive instructor feedback and identify changes needed for the next report stage.
Week 3–4	High	Update D&C report and website	Revise early documentation and upload the updated D&C report to the group website.

Week 4	High	Develop engineering specifications	Convert broad project goals into measurable specifications such as shot detection latency, app update delay, wireless range, battery life, and calibration behavior.
Week 4	Medium	Create House of Quality	Compare customer requirements against engineering characteristics and identify major design tradeoffs.
Week 4–5	High	Select major system architecture	Define the major hardware and software subsystems: sensor array, ESP32-S3 control module, LED/display feedback, BLE communication, mobile app, storage, power, and enclosure.
Week 5	High	Create hardware block diagram	Show how the sensor array, multiplexers, ESP32, display, buttons, power bank, regulators, and communication blocks connect.
Week 5	High	Create software flowchart	Show the main firmware path from startup to shot detection, scoring, display feedback, logging, and app communication.
Week 5–6	High	Research phototransistor and MUX options	Compare sensor and multiplexer options, confirm test components, and identify risks related to ambient light and analog signal routing.
Week 6	Medium	Define PCB design direction	Establish the target PCB size, sensor layout strategy, test points, connectors, power routing needs, and major layout constraints.
Week 6	Medium	Define power system direction	Decide on a commercial rechargeable power bank strategy and determine required voltage regulation for 5 V and 3.3 V rails.
Week 6–7	High	Prepare midterm report	Expand the report with hardware design, software design, prototype design, budget, and milestone planning.
Week 7	High	Attend midterm report group meeting	Receive feedback on the project plan, report quality, component choices, and implementation readiness.
Week 7–8	High	Update midterm report and website	Revise the midterm report, clean up diagrams, improve documentation, and upload the updated report to the project website.
Week 8	High	Create early firmware proof-of-concept	Begin basic ESP32-S3 firmware structure, including sensor test code, BLE planning, display planning, or serial debug output.
Week 8	High	Create early mobile app proof-of-concept	Scaffold the React Native app and prepare the basic screens needed for BLE scanning, connection status, shot history, and settings.
Week 8–9	Medium	Prepare mini demo concept	Decide what can be shown in the SD1 mini demo, such as app interface, ESP32 test output, basic sensor test, website, or system simulation.
Week 9	High	Finalize SD1 component decisions	Confirm the parts that must be ordered early in SD2, including ESP32-S3 module/dev kit, phototransistors, MUXs, LEDs, display, regulators, power source, and PCB-related parts.
Week 9–10	High	Complete SD1 final report	Finish the final technical report with expanded design reasoning, specifications, diagrams, budget, references, and AI usage disclosure.

Week 10	High	Complete SD1 mini demo video	Record a short demo showing project progress, planned system behavior, early tests, app progress, diagrams, or prototype concept.
Week 10	High	Submit SD1 final deliverables	Submit the SD1 final report and mini demo video, and ensure the project website contains the required documentation.

9.3 Senior Design 2 Milestones

Table 9.2: Senior Design 2 Milestones and Completion Criteria

Week	Priority	Milestone	Description / Completion Criteria
Week 1	High	Freeze final 500-channel system architecture	Confirm the 20-by-25 phototransistor array, 500-LED feedback array, multiplexer and ADC architecture, pulse-capture approach, ESP32-S3 controller, display, microSD storage, BLE communication, power system, and enclosure interfaces.
Week 1	High	Order remaining prototype components	Purchase the remaining ALS-PT19-315C/L177/TR8 sensors, IN-S126ATB LEDs, 74HC4067 multiplexers, LED drivers, passive components, TLV62569DBVT power components, INA219 monitor, connectors, PCB materials, enclosure hardware, and backup parts.
Week 1	High	Prepare hardware validation environment	Set up the ESP32-S3 development board, Mantis Pink Rhino cartridge, sensor test circuits, oscilloscope, logic analyzer, multimeter, bench power supply, test fixtures, and measurement procedures.
Week 1–2	High	Characterize laser and phototransistor response	Measure the training laser's pulse duration, sensor rise and fall times, direct-laser response, ambient-light baseline, signal amplitude, and repeatability under controlled indoor conditions.
Week 2	High	Validate multiplexer and ADC scan path	Connect a reduced group of sensors through the 74HC4067 network and measure channel selection, settling time, scan speed, ADC stability, noise, and signal degradation.
Week 2	High	Validate LED feedback architecture	Test the selected LED-driver approach, current-limiting resistors, brightness, response timing, addressing method, power consumption, and thermal behavior.
Week 2–3	High	Build reduced sensor and LED prototype	Construct a small sensor-and-LED grid with the mechanical isolation structure to test shot detection, weighted location estimation, optical crosstalk, ambient-light rejection, and localized feedback.
Week 3	High	Finalize ESP32-S3 resource allocation	Assign all required GPIO, ADC, SPI, I2C, USB, button, display, storage, multiplexer, LED-driver, battery-monitoring, and debugging connections.
Week 3	High	Finalize pulse-capture and scan-timing solution	Confirm that the complete sensor scan can capture the Pink Rhino laser pulse. Implement and validate optimized scanning, peak-hold, or pulse-latching hardware if the pulse is shorter than the scan period.
Week 3–4	High	Complete final PCB schematic	Finalize the schematic for the 500 sensors, 500 LEDs, multiplexers, LED drivers, ESP32-S3, display, touch controller, microSD storage, controls, USB-C input, TLV62569DBVT regulator, INA219 monitor, protection components, and test points.
Week 4	High	Perform schematic and manufacturing review	Verify component values, footprints, pin assignments, power limits, grounding, bypass capacitors, sensor mapping, protection circuits, test access, and manufacturer requirements before PCB release.
Week 4–5	High	Complete final PCB layout	Route the complete board while preserving sensor and LED alignment, analog-signal integrity, power distribution, antenna clearance, mounting points, test points, and enclosure compatibility.

Week 5	High	Release PCB and stencil for fabrication	Generate and review the Gerber files, drill files, fabrication notes, bill of materials, placement data, and stencil files before submitting the board for manufacturing.
Week 5–6	Medium	Complete firmware foundation during PCB fabrication	Finish drivers and modules for sensor scanning, calibration, shot processing, scoring, display control, physical controls, microSD logging, BLE communication, system states, and error reporting.
Week 5–6	Medium	Complete mobile-app foundation during PCB fabrication	Finish BLE scanning and connection, JSON packet parsing, shared application state, live dashboard, settings controls, SQLite storage, training history, analytics, and debugging views.
Week 6–7	High	Assemble and inspect first PCB revision	Apply solder paste, place and solder components, inspect polarity and orientation, check continuity, identify shorts or open connections, and document required rework.
Week 7	High	Bring up and validate the power system	Verify the 5 V input rail and TLV62569DBVT 3.3 V rail during startup, idle operation, display use, sensor scanning, BLE communication, and LED activity. Confirm INA219 measurements and protection behavior.
Week 7–8	High	Bring up digital peripherals and controls	Program the ESP32-S3 and verify USB Serial/JTAG, TFT display, capacitive touchscreen, microSD card, physical controls, BLE advertising, and basic system-state handling.
Week 8	High	Validate all 500 sensor channels	Confirm that every phototransistor can be selected, read, calibrated, mapped to the correct target coordinate, and identified as functional, noisy, weak, or disconnected.
Week 8–9	High	Validate all 500 LED channels	Confirm that every LED can be addressed at the correct target location with acceptable brightness, response time, current consumption, and thermal performance.
Week 9	High	Complete shot-detection and calibration firmware	Implement per-sensor baselines, global sensitivity control, threshold filtering, strongest-response detection, cluster grouping, weighted location estimation, outlier rejection, and duplicate-shot lockout.
Week 9	High	Complete scoring and training-mode firmware	Implement Freestyle, Classic, and Rapid modes, including shot count, score, session state, manual distance tracking, target feedback, and reset behavior.
Week 9–10	High	Finalize BLE protocol and state synchronization	Implement versioned JSON shot packets, target-state packets, control commands, reconnection behavior, state requests, error messages, and synchronization between the target and mobile application.
Week 10	Medium	Complete mobile history and analytics integration	Store shot and session records in SQLite and complete live shot display, history grouped by session and date, average score, performance by mode, performance by distance, grouping analysis, and debug information.
Week 9–10	High	Fabricate enclosure, faceplate, and optical grid	Produce the enclosure, 500-cell isolation grid, target faceplate, display opening, control openings, battery compartment, USB-C access, ventilation, mounting points, and serviceable rear panel.
Week 10–11	High	Install PCB and internal hardware	Mount the PCB, display, power bank, switches, controls, cables, and enclosure hardware while maintaining sensor-grid alignment, cable strain relief, antenna clearance, and access to test points.
Week 11	High	Complete full-system integration	Combine the sensor array, LED feedback, firmware, display, physical controls, microSD storage, power system, BLE communication, mobile application, and enclosure into one functioning prototype.
Week 11–12	High	Perform indoor-lighting calibration	Calibrate and test the system under dark, normal, and bright indoor lighting. Establish usable sensitivity settings and verify that lighting changes do not create excessive false detections.

Week 12	High	Verify demonstrable engineering specifications	Test at least 30 controlled shots for 90% detection within 2.0 cm, 95% onboard feedback within 500 ms, 95% mobile updates within one second at up to 10 m, and correct mapping of all 500 sensor channels.
Week 12–13	High	Perform reliability, power, and endurance testing	Run repeated sessions, four-hour battery testing, voltage-rail measurements, BLE disconnection tests, storage-failure tests, thermal checks, and mechanical inspections for loose or misaligned components.
Week 13–14	High	Correct integration issues and finalize prototype	Repair PCB faults, replace failed components, tune detection thresholds, correct mapping errors, improve firmware and app behavior, revise enclosure parts, and repeat any failed specification tests.
Week 14–15	High	Complete final validation and presentation preparation	Repeat the demonstrable tests, record final results, photograph the prototype, update the final budget, complete the report, prepare the presentation and video, and rehearse the complete demonstration.
Week 16	High	Complete final prototype demonstration	Demonstrate startup, lighting calibration, standalone controls, Freestyle, Classic, and Rapid modes, shot detection, location and scoring feedback, BLE communication, mobile history, and continued operation without a connected phone.

9.4 Priority-Based Development Summary

The team will use a priority-based development strategy to ensure that essential system functions are completed before time and resources are assigned to optional features. This approach reduces the risk that lower-value improvements delay the completion, testing, or demonstration of the core PrecisionShot prototype.

Table 9.3: Priority-Based Development Summary

Priority Level	Required Work
High Priority	Sensor detection, MUX scanning, ESP32 firmware, PCB design, power stability, display feedback, physical controls, BLE communication, calibration, enclosure alignment, and final demonstration readiness.
Medium Priority	Mobile app analytics, local app history, microSD logging, improved UI design, enclosure polish, expanded debug tools, and training mode refinement.
Low Priority	Automatic distance tracking, advanced coaching feedback, customizable drills, shot angle tracking, cloud backup, and additional app visualizations.

Development will begin with the high-priority tasks required for the system to operate as intended. These tasks establish the complete path from laser detection and sensor scanning to processing, feedback, wireless communication, and physical integration. The prototype will not be considered functionally complete until these areas operate reliably together.

Medium-priority work will begin after the related core subsystems have been implemented and validated. These features will improve usability, data storage, debugging, presentation quality, and training functionality without replacing the essential system behavior.

Low-priority features will only be attempted after the prototype meets its primary engineering requirements and performs consistently during testing. If schedule, budget, or technical limitations arise, these features may be deferred so the team can focus on reliability, final validation, and demonstration readiness.

9.5 Team Responsibilities

The PrecisionShot project is divided among four team members based on each member's primary technical focus and project-management role. Although each member has a defined area of responsibility, the system requires continuous coordination because the software, electronics, sensing hardware, power system, enclosure, and user interface must operate as one integrated platform. Team members will therefore support one another during design reviews, testing, troubleshooting, documentation, and final system integration.

DeLayne Russell will serve as the team lead and software design lead. His primary responsibility is the development and coordination of the software systems that control the target and companion mobile application. This includes the timer-based sensor-scanning process, multiplexer control logic, ambient-light calibration, shot-detection algorithm, sensor clustering, outlier rejection, shot-location estimation, scoring, training-mode behavior, and duplicate-shot prevention. He will also be responsible for the firmware that updates the onboard display, processes physical user inputs, manages system states, records local data, and handles software error conditions.

DeLayne will also lead the design and development of Bluetooth Low Energy communication between the ESP32-S3 and the mobile application. This includes defining the communication protocol, organizing JSON data packets, sending shot and system-status information, receiving remote-control commands, and managing connection, disconnection, and reconnection behavior. On the mobile side, he will oversee the React Native and TypeScript application, including the live dashboard, Bluetooth controls, local SQLite storage, shot history, user statistics, training analytics, settings, and debugging interfaces.

As team lead, DeLayne will coordinate task assignments, monitor deadlines, organize team progress, maintain the project milestone list, and help ensure that work completed by different members remains consistent with the overall system design. He will also assist with report organization, integration planning, demonstration preparation, and communication between the team and faculty reviewers.

Anthony Fontana will serve as the hardware design lead. His primary responsibility is the design and integration of the project's electronic hardware. This includes the 500-phototransistor sensing array, 500-LED feedback array, analog multiplexer network, LED-driver circuitry, sensor biasing, signal conditioning, and electrical routing between the sensor array and ESP32-S3 controller.

Anthony will also lead the schematic and PCB development process. His responsibilities include component placement, trace routing, power and ground distribution, connector selection, decoupling, test-point placement, and verification of component footprints. He will ensure that the sensor, LED, multiplexer, display, storage, power, and control circuits are connected correctly and that the final PCB can be assembled, tested, and repaired when necessary.

The power system will also fall under Anthony's primary responsibilities. This includes integrating the USB-C power input, KUXIU power bank, TLV62569DBVT buck regulator, INA219 current monitor, polyfuse, filtering components, and the 5 V and 3.3 V power rails. He will verify voltage stability, expected current capacity, regulator operation, electrical protection, and compatibility between the selected hardware components. During system integration, Anthony will lead hardware bring-up, electrical troubleshooting, oscilloscope measurements, continuity testing, and validation of sensor and multiplexer signals.

Kenn Pickavance will serve as the specifications and research lead. His responsibilities include defining and maintaining the project goals, objectives, engineering requirements, and demonstrable specifications. He will ensure that the project's performance targets are measurable, technically reasonable, and connected to an appropriate verification method. These specifications include shot-detection accuracy, response time, sensor coverage, ambient-light rejection, Bluetooth range, application update time, standalone operation, battery runtime, and data-storage requirements.

Kenn will also lead technical research for major component and technology decisions. This includes researching phototransistors, analog multiplexers, microcontrollers, displays, storage methods, wireless communication technologies, voltage regulators, battery systems, laser-training cartridges, and relevant signal-processing approaches. His work will include comparing candidate technologies, reviewing datasheets, documenting tradeoffs, developing comparison tables, and supporting the final selection of components using technical and cost-based evidence.

In addition, Kenn will be responsible for standards research, safety documentation, source collection, references, and validation planning. He will help identify applicable engineering, electrical, laser-safety, wireless, and design guidelines. During Senior Design II, he will assist with the development of formal test procedures and will help compare measured prototype performance against the specifications established in the report.

Nicolas "Nick" Koteff will serve as the prototype and enclosure lead. His primary responsibility is the physical design and construction of the PrecisionShot target. This includes the overall target dimensions, active sensing region, control-panel layout, rear enclosure, mounting system, faceplate, internal support structure, and access points for the display, buttons, USB-C connection, battery, and debugging hardware.

Nick will lead the mechanical integration between the enclosure and the electronic system. Each phototransistor and LED must remain correctly aligned with its corresponding faceplate opening, so his responsibilities include sensor-grid spacing, aperture dimensions, light-isolation walls, component clearance, PCB mounting points, screw locations, standoffs, threaded inserts, cable-routing paths, and tolerances between printed and electronic parts. He will also design the enclosure so that internal components remain protected while still being accessible for testing, firmware updates, repairs, and battery maintenance.

The enclosure must reduce unwanted ambient light while allowing direct laser light to reach the intended sensors. Nick will therefore be responsible for the grid-based optical isolation structure, diffuser or faceplate materials, surface finish, and alignment between the target graphics and sensing positions. He will also consider ventilation, structural rigidity, weight distribution, portability, assembly order, and resistance to movement or flexing during normal use.

Nick will oversee three-dimensional modeling, prototype printing, fit testing, enclosure revisions, and final mechanical assembly. He will coordinate closely with Anthony to confirm that the PCB and hardware fit the enclosure and with DeLayne to ensure that the display, buttons, and other controls are positioned appropriately for the software interface.

Although responsibilities are assigned by area, final integration and testing will be shared by the entire team. All members will participate in design reviews, subsystem testing, troubleshooting, documentation, prototype assembly, demonstration preparation, and verification of the final engineering specifications. This shared integration process will help ensure that the completed PrecisionShot system functions as a unified hardware, software, and mechanical design.

9.6 Final Deliverables

Table 9.4: Final Project Deliverables and Completion Criteria

Deliverable	Priority	Completion Criteria
SD1 final report	High	Complete technical report submitted by the SD1 deadline.
SD1 mini demo video	High	Short video showing project progress, system concept, and early prototype/demo work.
SD2 final report	High	Final report documenting completed design, testing, results, and prototype performance.
Working PrecisionShot prototype	High	Physical target detects laser shots and provides real-time feedback.
Functional phototransistor sensor array	High	Sensor array detects laser input and maps readings to target positions.
Functional LED or display feedback system	High	Target shows shot location, score, mode, and system status.
Working ESP32 firmware	High	Firmware handles sensor reading, shot detection, scoring, calibration, display, controls, and BLE.
Rechargeable power system	High	Target operates from a rechargeable power source with stable voltage rails.
Physical target enclosure	High	Enclosure protects electronics, aligns sensors, provides mounting, and allows user interaction.
Mobile app prototype	High	App connects to target, receives shot data, shows live status, and stores local history.
Calibration and testing results	High	Testing proves shot detection, response time, lighting calibration, wireless behavior, and system reliability.
Final presentation slides	High	Slides summarize problem, design, implementation, testing, and final results.
Final project demonstration	High	Team demonstrates the working system to the review committee.
Final project video	Medium	Video documents final system operation and major features.

Updated group website	Medium	Website includes current reports, diagrams, documentation, photos, and project updates.

Chapter 10: LLM-Assisted Development

10.1 Role of Large Language Models

Large language models were used throughout the development of the PrecisionShot Training System as research, planning, programming, troubleshooting, and documentation tools. Their purpose was not to replace engineering judgment, physical testing, or manufacturer documentation. Instead, LLMs helped the team move more quickly from an initial problem to a structured set of possible solutions that could be evaluated and verified.

PrecisionShot combines optical sensing, analog multiplexing, embedded firmware, Bluetooth communication, mobile application development, power regulation, PCB design, and mechanical enclosure construction. Each area requires different technical knowledge and introduces its own terminology, components, and design tradeoffs. LLM assistance made it easier to explore these subjects, identify relevant design questions, and organize information before performing deeper research.

The main benefit was reducing the time required to begin meaningful engineering work. Rather than starting every research task without direction, the team could describe a requirement and use an LLM to identify possible technologies, comparison criteria, implementation risks, and areas requiring additional verification.

10.2 Research and Component Comparisons

LLMs were especially useful during the research and comparison of major hardware components. They helped the team identify relevant selection criteria such as operating voltage, current consumption, response time, spectral sensitivity, package size, channel count, communication interface, availability, cost, and manufacturing difficulty.

For the optical sensing system, LLM-assisted research helped explain phototransistor behavior, ambient-light interference, spectral response, dark current, and analog signal generation. This supported comparisons between through-hole sensors used during testing and surface-mount phototransistors intended for the final PCB. It also helped the team evaluate how closely the selected phototransistor response matched the wavelength of the Mantis Pink Rhino laser-training cartridge.

LLMs also supported the comparison of analog multiplexers and ADC architectures. Candidate devices such as the 74HC4067, ADG1606, and ADG732 were compared according to channel count, switching speed, on-resistance, cost, and signal quality. Discussions of single-stage and cascaded multiplexing helped the team understand the relationship between sensor quantity, ESP32-S3 input limitations, scan time, and possible signal-settling delays.

The same process was used when evaluating voltage regulators, displays, LED packages, switches, storage methods, wireless technologies, and battery-monitoring components. LLMs helped create organized comparison tables and identify tradeoffs, while the team verified final specifications through datasheets, supplier information, calculations, and physical testing.

This approach allowed the team to evaluate more alternatives within the available Senior Design schedule. It also improved the quality of the report by supporting component selections with documented comparisons rather than unsupported preferences.

10.3 Software and System Design Support

LLMs were used to support development of both the ESP32-S3 firmware and the companion mobile application. For the embedded system, they helped divide the firmware into manageable responsibilities, including initialization, multiplexer scanning, sensor calibration, shot detection, scoring, display control, local logging, Bluetooth communication, and error handling.

The shot-detection architecture was refined through repeated LLM-assisted discussions. Early concepts relied mainly on a fixed detection threshold. Further analysis led to the use of an individual ambient-light baseline for each sensor combined with one global sensitivity setting. The detection process was also expanded to identify the strongest response, group nearby sensors, reject distant outliers, estimate the shot location, and prevent one laser pulse from being recorded multiple times.

LLMs also helped identify edge cases such as weak signals, reflections, changing room lighting, multiple independent sensor clusters, and laser pulses shorter than the full sensor scan time. These discussions helped the team plan testing and supporting hardware before completing the full array.

For the mobile application, LLM assistance supported the separation of the interface, Bluetooth communication, shared application state, SQLite database, and analytics logic. It also helped organize screens for connection management, live shot feedback, settings, training history, and debugging. During implementation, LLMs may also be used to interpret build errors, review TypeScript code, and suggest troubleshooting procedures.

Hardware and mechanical design discussions benefited as well. LLMs helped connect PCB requirements, sensor alignment, optical isolation, power distribution, display placement, wiring access, and enclosure construction into one coordinated system design.

10.4 Effect on Project Scope and Innovation

The use of LLMs significantly increased the scope that the team could reasonably investigate within the Senior Design timeline. Without this assistance, the project would likely have required a simpler architecture with fewer component comparisons, less detailed software planning, and fewer advanced features.

Faster research and development gave the team greater confidence when evaluating unfamiliar technologies. This encouraged the consideration of a dense sensor and LED array, timer-controlled multiplexing, individual sensor calibration, cluster-based shot-location estimation, onboard scoring, BLE communication, local mobile storage, and multiple training modes. LLMs did not make these features automatically successful, but they reduced the time required to understand the problems and develop initial implementation plans.

This created an environment in which the team was more willing to investigate ambitious milestones rather than immediately removing them because of limited research time. Some ideas may still be reduced or postponed after testing, but those decisions can be made using stronger technical reasoning.

LLM assistance may remain valuable during Senior Design II and future development. It can help the team interpret test results, compare design revisions, organize debugging information, improve software architecture, and investigate additional training features. In this way, LLMs support innovation by allowing more engineering possibilities to be explored before time and budget limitations require a final decision.

10.5 Verification and Responsible Use

LLM output was not treated as an authoritative engineering source. Large language models can provide incorrect specifications, confuse similar components, produce faulty calculations, or present assumptions as confirmed facts. For this reason, all important technical information was independently verified.

Component specifications are checked against manufacturer datasheets and supplier documentation. Electrical behavior is confirmed through calculations and physical measurements. Firmware and mobile application code must be reviewed, compiled, and tested in the actual development environment. Safety-related information must be supported by appropriate manufacturer or regulatory sources.

LLMs were also used to improve report organization and writing, but the team remained responsible for ensuring that every section accurately reflected the current design. Generated text was reviewed for incorrect quantities, outdated parts, unsupported claims, and contradictions between chapters.

Overall, LLMs served as powerful engineering-assistance tools that accelerated research, strengthened comparisons, supported development, and expanded the project's practical scope. Final design decisions, verification, testing, and responsibility for the completed PrecisionShot system remained with the team.

Chapter 11: Conclusion

The PrecisionShot Training System was developed to provide a safer, more accessible, and more informative alternative to traditional live-fire practice. The project expands beyond a basic hit-detection target by combining a dense phototransistor sensing array, localized LED feedback, an onboard display, embedded scoring, rechargeable power, local storage, and Bluetooth communication with a companion mobile application. Together, these features are intended to provide immediate shot feedback while also allowing users to review training history, evaluate accuracy, and track improvement over time.

During Senior Design I, the team established the project requirements and completed the major research, architecture, and planning needed to begin prototype construction. The hardware design defines how the phototransistor array, analog multiplexers, ESP32-S3 controller, LED drivers, display, microSD storage, physical controls, power system, and supporting circuitry will operate as one integrated system. Component comparisons were completed for the major sensing, processing, display, power, wireless, and laser subsystems so that selections could be based on measurable engineering criteria rather than convenience alone. Particular attention was given to ambient-light rejection, sensor scanning speed, analog signal quality, power efficiency, mechanical alignment, and laser safety because these factors directly affect the reliability of the final target.

The software architecture was designed around a device-first approach in which the embedded target remains fully functional without a connected phone. The ESP32-S3 will perform sensor scanning, lighting calibration, shot detection, location estimation, scoring, mode control, display updates, local logging, and Bluetooth communication. The mobile application will extend the system with remote controls, local SQLite storage, training history, statistics, and diagnostic tools. Separating the real-time target functions from the higher-level mobile features improves reliability while still providing the benefits of a modern connected training platform.

The physical prototype design was developed to support both the electronic and optical requirements of the system. The enclosure must protect the electronics, maintain alignment between the sensors, LEDs, PCB, and target face, reduce interference from off-axis ambient light, and provide convenient access to the display, controls, USB-C connection, and internal hardware. The grid-based optical isolation structure is especially important because the system must distinguish short laser pulses from normal indoor lighting. The mechanical and electrical designs were therefore developed together rather than as independent parts of the project.

The current design represents an ambitious but achievable Senior Design prototype. The estimated budget, milestone schedule, subsystem responsibilities, testing procedures, and risk-control strategies provide a practical plan for completing the system during Senior Design II. Development will begin with small-scale validation of the sensors, multiplexers, power system, display, firmware, and mobile application before progressing to PCB fabrication, enclosure construction, full integration, and formal performance testing. Accuracy, response time, lighting reliability, Bluetooth operation, battery runtime, storage, and standalone functionality will be measured against the demonstrable specifications established in this report.

PrecisionShot has the potential to provide meaningful training feedback without requiring live ammunition, a shooting range, an external camera, or a constant internet connection. The success of the project will ultimately depend on careful subsystem testing and the ability to integrate the optical, electrical, software, and mechanical designs into one reliable device. Senior Design I established the technical foundation for that work. Senior Design II will focus on converting the completed research and design into a functional prototype, validating its performance, correcting weaknesses discovered during testing, and demonstrating that the PrecisionShot concept can operate as a practical laser-based training system.

Appendix A: References

[1] LaserLyte, “Laser Training Solutions,” LaserLyte. Accessed: Jul. 5, 2026. [Online]. Available: https://www.laserlyte.com/

[2] MantisX, “BlackbeardX: Dry Fire Laser Training System with Smart Analytics + Feedback,” Mantis. Accessed: Jul. 5, 2026. [Online]. Available: https://mantisx.com/products/blackbeardx-the-auto-resetting-trigger-system-with-analytics-and-smart-feedback

[3] Espressif Systems, “ESP32-S3-WROOM-1 and ESP32-S3-WROOM-1U Datasheet,” Espressif Systems. Accessed: Jul. 5, 2026. [Online]. Available: https://documentation.espressif.com/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf

[4] DigiKey, “ESP32-S3-WROOM-1-N8R8 Espressif Systems RF Transceiver Module Product Page,” DigiKey. Accessed: Jul. 5, 2026. [Online]. Available: https://www.digikey.com/en/products/detail/espressif-systems/ESP32-S3-WROOM-1-N8R8/15295891

[5] Espressif Systems, “ESP-IDF Programming Guide,” Espressif Systems. Accessed: Jul. 5, 2026. [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/stable/esp32/index.html

[6] Texas Instruments, “LMR33620 SIMPLE SWITCHER 3.8-V to 36-V, 2-A Synchronous Step-Down Voltage Converter Datasheet,” Texas Instruments. Accessed: Jul. 5, 2026. [Online]. Available: https://www.ti.com/lit/ds/symlink/lmr33620.pdf

[7] DigiKey, “LMR33620ADDA Texas Instruments DC-DC Switching Regulator Product Page,” DigiKey. Accessed: Jul. 5, 2026. [Online]. Available: https://www.digikey.com/en/products/detail/texas-instruments/LMR33620ADDA/8571546

[8] Texas Instruments, “CD74HCx4067 High-Speed CMOS Logic 16-Channel Analog Multiplexer and Demultiplexer Datasheet,” Texas Instruments. Accessed: Jul. 5, 2026. [Online]. Available: https://www.ti.com/lit/ds/symlink/cd74hc4067.pdf

[9] DigiKey, “CD74HC4067M96 Texas Instruments Analog Multiplexer Product Page,” DigiKey. Accessed: Jul. 5, 2026. [Online]. Available: https://www.digikey.com/en/products/detail/texas-instruments/CD74HC4067M96/1507236

[10] Everlight Electronics, “ALS-PT19-315C/L177/TR8 Ambient Light Sensor Datasheet,” Everlight Electronics. Accessed: Jul. 5, 2026. [Online]. Available: https://www.everlighteurope.com/custom/files/datasheets/DLS-0000039.pdf

[11] DigiKey, “ALS-PT19-315C/L177/TR8 Everlight Electronics Phototransistor Product Page,” DigiKey. Accessed: Jul. 5, 2026. [Online]. Available: https://www.digikey.com/en/products/detail/everlight-electronics-co-ltd/ALS-PT19-315C-L177-TR8/2675735

[12] Everlight Electronics, “ALS-PT204-6C/L177 Ambient Light Sensor 3 mm T-1 Datasheet,” Everlight Electronics. Accessed: Jul. 5, 2026. [Online]. Available: https://www.everlighteurope.com/custom/files/datasheets/DLS-0000030.pdf

[13] DigiKey, "ALS-PT204-6C/L177 Everlight Electronics Phototransistor Product Page," DigiKey. Accessed: Jul. 5, 2026. [Online]. Available: https://www.digikey.com/en/products/detail/everlight-electronics-co-ltd/ALS-PT204-6C-L177/2675510

[14] Vishay Semiconductors, "TEPT5700 Ambient Light Sensor Datasheet," Vishay. Accessed: Jul. 5, 2026. [Online]. Available: https://www.vishay.com/docs/81321/tept5700.pdf

[15] Vishay Semiconductors, "TEPT4400 Ambient Light Sensor Datasheet," Vishay. Accessed: Jul. 5, 2026. [Online]. Available: https://www.vishay.com/docs/81341/tept4400.pdf

[16] Würth Elektronik, "150080VS75000 WL-SMCW SMT Mono-Color Chip LED Datasheet," Würth Elektronik. Accessed: Jul. 5, 2026. [Online]. Available: https://www.we-online.com/components/products/datasheet/150080VS75000.pdf

[17] DigiKey, "150080VS75000 Würth Elektronik 0805 Green LED Product Page," DigiKey. Accessed: Jul. 5, 2026. [Online]. Available: https://www.digikey.com/en/products/detail/w%C3%BCrth-elektronik/150080VS75000/4489924

[18] DigiKey, "LEDDC-5CARGB Lumimax Optoelectronic Technology RGB LED Product Page," DigiKey. Accessed: Jul. 5, 2026. [Online]. Available: https://www.digikey.com/en/products/detail/lumimax-optoelectronic-technology/LEDDC-5CARGB/29263561

[19] Bluetooth SIG, "The Bluetooth Low Energy Primer," Bluetooth Special Interest Group. Accessed: Jul. 5, 2026. [Online]. Available: https://www.bluetooth.com/bluetooth-le-primer/

[20] Android Developers, "Bluetooth Low Energy Overview," Google Android Developers. Accessed: Jul. 5, 2026. [Online]. Available: https://developer.android.com/develop/connectivity/bluetooth/ble/ble-overview

[21] React Native, "Get Started with React Native," Meta Open Source. Accessed: Jul. 5, 2026. [Online]. Available: https://reactnative.dev/docs/environment-setup

[22] Expo, "Expo Documentation," Expo. Accessed: Jul. 5, 2026. [Online]. Available: https://docs.expo.dev/

[23] PCBWay, "PCB Prototype and Fabrication Manufacturer," PCBWay. Accessed: Jul. 5, 2026. [Online]. Available: https://www.pcbway.com/

[24] PCBWay, "How to Place a SMD-Stencil Order at PCBWay," PCBWay. Accessed: Jul. 5, 2026. [Online]. Available: https://www.pcbway.com/blog/help_center/How_to_Place_a_SMD_Stencil_Order_at_PCBWay_PCBWay_Website_Exploration_03_50f4e750.html

[25] KUXIU, "KUXIU Official Store," KUXIU. Accessed: Jul. 5, 2026. [Online]. Available: https://kuxiu.co/

[26] A. Hollister, "Kuxiu's 'world first' solid-state power bank costs more but lasts much longer," The Verge. Accessed: Jul. 5, 2026. [Online]. Available: https://www.theverge.com/reviews/656183/kuxius-semi-solid-state-power-bank-costs-a-little-more-but-lasts-much-longer

[27] A. Hollister, “Kuxiu’s S3 MagSafe power bank is bigger and better and probably won’t explode,” The Verge. Accessed: Jul. 5, 2026. [Online]. Available: https://www.theverge.com/reviews/846094/kuxiu-s3-solid-state-power-bank-review

[28] YouTube, “Arduino To ESP32: How to Get Started!” Accessed: Jul. 5, 2026. [Online]. Available: https://www.youtube.com/watch?v=RiYnucfy_rs

[29] YouTube, “Introduction to ESP32 - Getting Started.” Accessed: Jul. 5, 2026. [Online]. Available: https://www.youtube.com/watch?v=xPlN_Tk3VLQ

[30] YouTube, “Getting Started with ESP32 S3 | Beginner’s Guide to Programming & Projects.” Accessed: Jul. 5, 2026. [Online]. Available: https://www.youtube.com/watch?v=QyFvK8ywm3Q

[31] YouTube, “JC4827W543 ESP32-S3 Board Full Tutorial.” Accessed: Jul. 5, 2026. [Online]. Available: https://www.youtube.com/watch?v=mnOzfRFQJIM

[32] YouTube, “Add a Touchscreen to Your ESP32 Project (It’s Easier Than You Think).” Accessed: Jul. 5, 2026. [Online]. Available: https://www.youtube.com/watch?v=GGGEvKVZpJQ

[33] Google, “Gemini,” Google AI. Accessed: Jul. 5, 2026. [Online]. Available: https://gemini.google.com/

[34] OpenAI, “ChatGPT,” OpenAI. Accessed: Jul. 5, 2026. [Online]. Available: https://chatgpt.com/

[35] Anthropic, “Claude,” Anthropic. Accessed: Jul. 5, 2026. [Online]. Available: https://claude.ai/

[36] GitHub, “GitHub Copilot,” GitHub. Accessed: Jul. 5, 2026. [Online]. Available: https://github.com/features/copilot

[37] Sitronix Technology Co., Ltd., "ST7796S: 320RGB x 480 dot 262K Color with Frame Memory Single-Chip TFT Controller/Driver Datasheet," Sitronix Technology. Accessed: Jul. 20, 2026. [Online]. Available: https://www.displayfuture.com/Display/datasheet/controller/ST7796s.pdf

[38] FocalTech Systems Co., Ltd., "FT6336U Self-Capacitive Touch Panel Controller Datasheet," FocalTech Systems. Accessed: Jul. 20, 2026. [Online]. Available: https://cdn-shop.adafruit.com/product-files/5826/FT6336U-DataSheet-V1.1.pdf

[39] Hosyond, "Hosyond 4.0 Inch 320x480 TN Capacitive Touch Screen LCD Display Module User Manual," manuals.plus. Accessed: Jul. 20, 2026. [Online]. Available: https://manuals.plus/asin/B0CRGQN58D

[40] Amazon, "Hosyond 4.0 Inch 320x480 TN Capacitive Touch Screen LCD Display Module SPI Serial ST7796S Driver for Arduino R3/Mega2560/ESP32 Product Page," Amazon.com. Accessed: Jul. 20, 2026. [Online]. Available: https://www.amazon.com/Hosyond-320x480-Capacitive-Display-Mega2560/dp/B0CRGQN58D

[41] Blikai, "Phototransistor: Circuit Pinout & Principle," Blikai Blog. Accessed: Jul. 26, 2026. [Online]. Available: https://www.blikai.com/blog/phototransistor-circuit-pinout-principle

[42] R. Paschotta, "Phototransistors," RP Photonics Encyclopedia. Accessed: Jul. 26, 2026. [Online]. Available: https://www.rp-photonics.com/phototransistors.html

[43] ScienceDirect, "Phototransistor," ScienceDirect Topics. Accessed: Jul. 26, 2026. [Online]. Available: https://www.sciencedirect.com/topics/engineering/phototransistor

[44] Electronics-Tutorials (AspenCore), "Phototransistor Basics," Electronics Tutorials. Accessed: Jul. 26, 2026. [Online]. Available: https://www.electronics-tutorials.ws/transistor/phototransistor-basics.html

[45] G. Schorcht, "ESP32-S3 Family," RIOT-OS API Documentation. Accessed: Jul. 26, 2026. [Online]. Available: https://api.riot-os.org/group__cpu__esp32__esp32s3.html

[46] Espressif Systems, ESP32-S3 Series Datasheet, Version 1.6. Accessed: Jul. 26, 2026. [Online]. Available: https://files.seeedstudio.com/wiki/SeeedStudio-XIAO-ESP32S3/res/esp32-s3_datasheet.pdf

[47] M. Oljaca and K. Sanborn, "Determining Minimum Acquisition Times for SAR ADCs When a Step Function Is Applied to the Input," Texas Instruments Application Report SBAA173A. Accessed: Jul. 26, 2026. [Online]. Available: https://www.ti.com/lit/pdf/sbaa173

[48] A. Walsh, "Front-End Amplifier and RC Filter Design for a Precision SAR Analog-to-Digital Converter," Analog Dialogue, Analog Devices. Accessed: Jul. 26, 2026. [Online]. Available: https://www.analog.com/en/resources/analog-dialogue/articles/front-end-amp-and-rc-filter-design.html

[49] Espressif Systems, "Analog to Digital Converter (ADC) — ESP32-S3," ESP-IDF Programming Guide v4.4.5. Accessed: Jul. 26, 2026. [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/v4.4.5/esp32s3/api-reference/peripherals/adc.html

[50] Espressif Systems, "ADC," Arduino-ESP32 Documentation. Accessed: Jul. 26, 2026. [Online]. Available: https://espressif-docs.readthedocs-hosted.com/projects/arduino-esp32/en/latest/api/adc.html

[51] ESP32 Developer, ".AtoD – One Shot Mode," ESP32 Developer. Accessed: Jul. 26, 2026. [Online]. Available: https://esp32developer.com/programming-in-c-c/atod/atod-one-shot-mode

[52] D. Taylor (stg), "ESP32-S3-FastAnalogRead," GitHub repository. Accessed: Jul. 26, 2026. [Online]. Available: https://github.com/stg/ESP32-S3-FastAnalogRead

[53] Espressif Systems, "ESP32 Forum, Topic No. 30102," ESP32 Forum. Accessed: Jul. 26, 2026. [Online]. Available: https://esp32.com/viewtopic.php?t=30102

[54] Espressif Systems, "GPIO & RTC GPIO — ESP32-S3," ESP-IDF Programming Guide. Accessed: Jul. 26, 2026. [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/stable/esp32s3/api-reference/peripherals/gpio.html

[55] Curious Scientist, "ESP32-S3 Multiple SPI Buses – SPI, VSPI, HSPI, FSPI, WTF…!?," Curious Scientist Blog. Accessed: Jul. 26, 2026. [Online]. Available: https://curiousscientist.tech/blog/esp32-s3-multiple-spi-buses-spi-fspi-hspi

[56] Texas Instruments, "ADS7953 Data Sheet, Product Information and Support," TI.com. Accessed: Jul. 26, 2026. [Online]. Available: https://www.ti.com/product/ADS7953

[57] Texas Instruments, ADS79xx Pin-Compatible, 12-, 10-, 8-Bit, 1-MSPS, 16-, 12-, 8-, 4-Channel, Single-Ended, Serial-Interface ADCs Datasheet, Rev. C. Accessed: Jul. 26, 2026. [Online]. Available: https://www.ti.com/lit/ds/symlink/ads7953.pdf

[58] Analog Devices, "MAX11128 Datasheet and Product Info," Analog Devices. Accessed: Jul. 26, 2026. [Online]. Available: https://www.analog.com/en/products/max11128.html

[59] Maxim Integrated (Analog Devices), MAX11120–MAX11128: 1Msps, Low-Power, Serial 12-/10-/8-Bit, 4-/8-/16-Channel ADCs Datasheet, Rev. 3. Accessed: Jul. 26, 2026. [Online]. Available: https://www.farnell.com/datasheets/2000877.pdf

[60] Microchip Technology Inc., "MCP3208," Microchip Product Page. Accessed: Jul. 26, 2026. [Online]. Available: https://www.microchip.com/en-us/product/mcp3208

[61] Microchip Technology Inc., "Sampling Rate and SPI Clock Frequency of MCP3204/MCP3208," Microchip Support Knowledge Base. Accessed: Jul. 26, 2026. [Online]. Available: https://support.microchip.com/s/article/Sampling-rate-and-SPI-Clock-frequency-of-MCP3204-MCP3208

[62] Circuit Globe, "Difference Between Photodiode & Phototransistor (with Comparison Chart)," Circuit Globe. Accessed: Jul. 26, 2026. [Online]. Available: https://circuitglobe.com/difference-between-photodiode-and-phototransistor.html

[63] JLCPCB, "Photodiode vs Phototransistor," JLCPCB Blog. Accessed: Jul. 26, 2026. [Online]. Available: https://jlcpcb.com/blog/photodiode-vs-phototransistor

[64] Zbotic, "LDR Photoresistor vs Phototransistor: Light Sensing Compared," Zbotic. Accessed: Jul. 26, 2026. [Online]. Available: https://zbotic.in/ldr-photoresistor-vs-phototransistor-light-sensing-compared-for-arduino-projects/

[65] R. Paschotta, "Position-Sensitive Detectors," RP Photonics Encyclopedia. Accessed: Jul. 26, 2026. [Online]. Available: https://www.rp-photonics.com/position_sensitive_detectors.html

[66] "CMOS-Compatible Position-Sensitive Devices (PSDs) Based on Photodetector Arrays," Sensors and Actuators A: Physical, ScienceDirect. Accessed: Jul. 26, 2026. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0924424703001262

[67] "Laser Spot Measurement Using Cost-Affordable Devices," arXiv:2004.13682. Accessed: Jul. 26, 2026. [Online]. Available: https://arxiv.org/pdf/2004.13682

[68] Everlight Electronics Co., Ltd., ALS-PT204-6C/L177 Ambient Light Sensor Datasheet, Document No. DAS-020-005 Rev. 1.1. Accessed: Jul. 26, 2026. [Online]. Available: https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/885/ALS-PT204-6C,L177.pdf

[69] Texas Instruments, Transimpedance Considerations for High-Speed Amplifiers, Application Report SBOA122. Accessed: Jul. 26, 2026. [Online]. Available: https://www.ti.com/lit/pdf/sboa122

[70] RP Photonics, "Photodiode Amplifiers," RP Photonics Encyclopedia. Accessed: Jul. 26, 2026. [Online]. Available: https://www.rp-photonics.com/photodiode_amplifiers.html

[71] ERSA Electronics, "Transimpedance Amplifier (TIA): Op-Amp Circuit, Design and ICs," ERSA Electronics Blog. Accessed: Jul. 26, 2026. [Online]. Available: https://www.ersaelectronics.com/blog/transimpedance-amplifier

[72] Texas Instruments, Comparator with Hysteresis Reference Design, TIDU020. Accessed: Jul. 26, 2026. [Online]. Available: https://www.ti.com/lit/pdf/tidu020

[73] Analog Devices, "Curing Comparator Instability with Hysteresis," Analog Dialogue. Accessed: Jul. 26, 2026. [Online]. Available: https://www.analog.com/en/resources/analog-dialogue/articles/curing-comparator-instability-with-hysteresis.html

[74] ScienceDirect, "Moving Average Filter," ScienceDirect Topics. Accessed: Jul. 26, 2026. [Online]. Available: https://www.sciencedirect.com/topics/engineering/moving-average-filter

[75] T. Roelandts, "The Moving Average as a Filter," Tom Roelandts. Accessed: Jul. 26, 2026. [Online]. Available: https://tomroelandts.com/articles/the-moving-average-as-a-filter

[76] Espressif Systems, ESP32-WROOM-32 Datasheet, Version 2.5. Accessed: Jul. 26, 2026. [Online]. Available: https://cdn-shop.adafruit.com/product-files/3320/3320_module_datasheet.pdf

[77] Espressif Systems, ESP32-WROOM-32 (ESP-WROOM-32) Datasheet, Version 2.4. Accessed: Jul. 26, 2026. [Online]. Available: https://www.mouser.com/datasheet/2/891/esp-wroom-32_datasheet_en-1223836.pdf

[78] ESPBoards, "ESP32-WROOM Development Board Details, Pinout, Specs," ESPBoards. Accessed: Jul. 26, 2026. [Online]. Available: https://www.espboards.dev/esp32/esp32-wroom/

[79] K. Kumar, "ESP32-WROOM-32: Features, Applications & Everything You Need to Know," ElectronifyIndia Blog. Accessed: Jul. 26, 2026. [Online]. Available: https://www.electronifyindia.com/blogs/news/esp32-wroom-32-features-applications-everything-you-need-to-know

[80] R. Mischianti, "STM32F103C8T6 Blue Pill: High Resolution Pinout and Specs," Mischianti. Accessed: Jul. 26, 2026. [Online]. Available: https://mischianti.org/stm32f103c8t6-blue-pill-high-resolution-pinout-and-specs/

[81] Fly-Wing Technology, "STM32F103C8T6 Blue Pill: Specs, Pinout, and Comparison," Fly-Wing Blog. Accessed: Jul. 26, 2026. [Online]. Available: https://www.flywing-tech.com/blog/stm32f103c8t6-blue-pill-specs-pinout-and-comparison/

[82] Xecor, "STM32F103C8T6 Blue Pill Development Board: Pinout, Specs, Programming & Datasheet," Xecor Blog. Accessed: Jul. 26, 2026. [Online]. Available: https://www.xecor.com/blog/stm32f103c8t6-blue-pill-pinout-specs-programming-datasheet

[83] MicrocontrollersLab, "STM32F103C8T6 Blue Pill: Pinout, Peripherals, Programming and Features," MicrocontrollersLab. Accessed: Jul. 26, 2026. [Online]. Available: https://microcontrollerslab.com/stm32f103c8t6-blue-pill-pinout-peripherals-programming-features/

[84] Raspberry Pi Ltd, "Buy a Raspberry Pi Pico," Raspberry Pi. Accessed: Jul. 26, 2026. [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-pico/

[85] Micro Center, "Raspberry Pi Pico W; RP2040 32-bit Cortex M0+ Dual Core Processor; 264KB RAM," Micro Center. Accessed: Jul. 26, 2026. [Online]. Available: https://www.microcenter.com/product/650108/raspberry-pi-pico-w

[86] Adafruit Industries, "Raspberry Pi Pico RP2040," Adafruit. Accessed: Jul. 26, 2026. [Online]. Available: https://www.adafruit.com/product/4864

[87] "RP2040 Ultra Low Power Wake on RTC," Raspberry Pi Forums. Accessed: Jul. 26, 2026. [Online]. Available: https://forums.raspberrypi.com/viewtopic.php?t=338245

[88] Strikeman, "Strikeman Dry-Fire Training System." Accessed Jul. 2026. https://www.strikeman.io/

[89] LaserHIT, "LaserHIT Laser Training Cartridge." Accessed Jul. 2026. https://www.laserhit.com/products/lasercartridge

[90] Mantis, "Laser Academy and Pink Rhino Laser Training Cartridge." Accessed Jul. 2026. https://mantisx.com/pages/laser-academy

[91] VNSH, "Digital Steel Laser Target Elite." Accessed Jul. 2026. https://vnsh.com/products/vnsh-digital-steel-laser-target-elite

[92] MidwayUSA, "Laser Activated Target." Accessed Jul. 2026. https://ads.midwayusa.com/product/1018119112

[93] IEEE TechNavigator, "Phototransistors," IEEE. Accessed Jul. 2026. [Online]. Available: https://technav.ieee.org/topic/phototransistors/

[94] ScienceDirect, "Planck Constant," Elsevier. Accessed Jul. 2026. [Online]. Available: https://www.sciencedirect.com/topics/engineering/planck-constant

[95] Texas Instruments, "Linear and Switching Voltage Regulator Fundamentals," Texas Instruments. Accessed Jul. 2026. [Online]. Available: https://www.ti.com/lit/pdf/snva558

[96] Analog Devices, "Advantages of Switching Regulators Over Linear Regulation," Analog Devices. Accessed Jul. 2026. [Online]. Available: https://www.analog.com/en/resources/technical-articles/advantages-of-switching-regulators-over-linear-regulation.html

[95] Texas Instruments, "TLV62569 2-A High Efficiency Step-Down Converter in SOT23 and 2-mm × 2-mm QFN Package," Datasheet.

[96] Diodes Incorporated, "AP63200/AP63201/AP63203 Synchronous Buck Converter," Datasheet.

[97] Texas Instruments, "LM2596 SIMPLE SWITCHER® Power Converter 150-kHz 3-A Step-Down Voltage Regulator," Datasheet.

[98] Advanced Monolithic Systems, "AMS1117 1-A Low Dropout Voltage Regulator," Datasheet.

[99] S. Ling, J. Sanny, and W. Moebs, "8.6 Lasers," in University Physics Volume 3. Houston, TX, USA: OpenStax, Rice University. Accessed: Jul. 2026. [Online]. Available: https://openstax.org/books/university-physics-volume-3/pages/8-6-lasers

[100] Coherent Corp., "What is a Laser?" Coherent. Accessed: Jul. 2026. [Online]. Available: https://www.coherent.com/news/glossary/laser

[101] ScienceDirect Topics, "Semiconductor Laser." Elsevier. Accessed: Jul. 2026. [Online]. Available: https://www.sciencedirect.com/topics/physics-and-astronomy/semiconductor-laser

[102] IEEE TechNavigator, "Phototransistors." Institute of Electrical and Electronics Engineers (IEEE). Accessed: Jul. 2026. [Online]. Available: https://technav.ieee.org/topic/phototransistors/

[103] ScienceDirect Topics, "Planck Constant." Elsevier. Accessed: Jul. 2026. [Online]. Available: https://www.sciencedirect.com/topics/engineering/planck-constant

[104] Strikeman, "Strikeman Laser Cartridge." Accessed: Jul. 2026. [Online]. Available: https://www.strikeman.io/products/strikeman-laser-cartridge

[105] LaserHIT, "LaserHIT Laser Training Cartridge." Accessed: Jul. 2026. [Online]. Available: https://www.laserhit.com/products/lasercartridge

[106] AeroLasec, "9 mm Dry-Fire Laser Training Cartridge." Accessed: Jul. 2026. [Online]. Available: https://aerolastec.com/products/dry-fire-laser-training-cartridge-9mm

[107] Mantis, "Pink Rhino Laser Training Cartridge – 9 mm." Accessed: Jul. 2026. [Online]. Available: https://mantisx.com/products/pink-rhino-laser-training-cartridge-9mm

[108] U.S. Food and Drug Administration, "Laser Products and Instruments," FDA. Accessed: Jul. 26, 2026. FDA Laser Products and Instruments

[109] U.S. Food and Drug Administration, "Frequently Asked Questions About Lasers," FDA. Accessed: Jul. 26, 2026. FDA Frequently Asked Questions About Lasers

[110] Occupational Safety and Health Administration, "Laser Hazards," OSHA. Accessed: Jul. 26, 2026. OSHA Laser Hazards

[111] PCBWay, "Board Size, Ordering Parameter Instruction," PCBWay Help Center. Accessed: Jul. 28, 2026. [Online]. Available: https://www.pcbway.com/helpcenter/ordering_parameter_instruction/Board_Size.html

[112] PCBWay, "PCB Assembly Capabilities," PCBWay. Accessed: Jul. 28, 2026. [Online]. Available: https://www.pcbway.com/assembly-capabilities.html

Appendix B: Copyright Permissions and AI Usage

All third-party images, diagrams, product information, datasheets, standards, and technical references used in this report are credited to their original authors, manufacturers, or organizations. Whenever possible, the team created original diagrams, tables, photographs, renderings, and technical explanations rather than reproducing copyrighted material. Any externally sourced content retained in the final report will be properly cited and used only for academic and educational purposes.

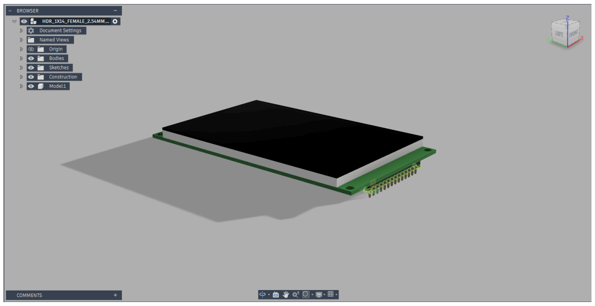
Product names, company names, trademarks, and logos referenced throughout the report remain the property of their respective owners. Their inclusion is intended only to identify and compare relevant technologies, components, and commercial products. The PrecisionShot team does not claim ownership of any third-party intellectual property.

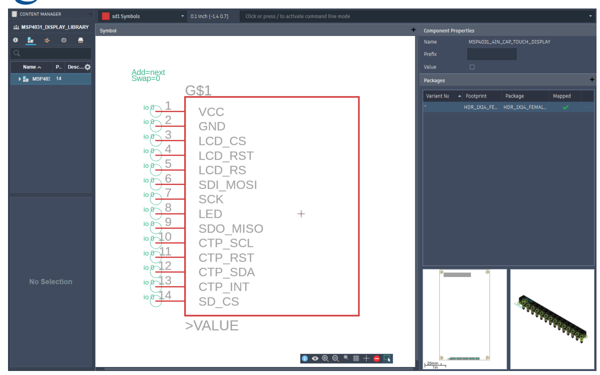
Generative AI tools, including ChatGPT, Google Gemini, Claude, and GitHub Copilot, were used as development aids for brainstorming, research organization, writing improvement, code troubleshooting, and early concept visualization. Their use is disclosed in accordance with the course AI policy.

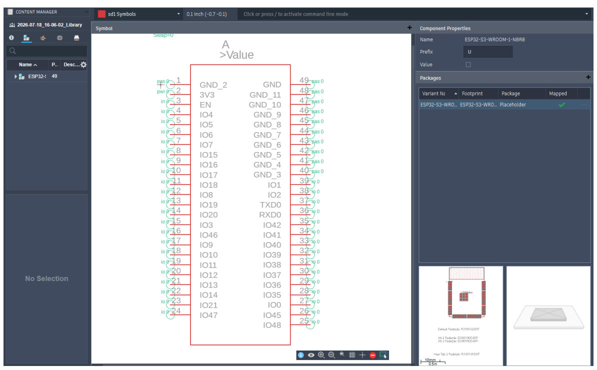
AI tools did not replace final engineering judgment, physical testing, calculations, or team review. All component selections, design decisions, software implementation, test results, and technical conclusions remain the responsibility of the PrecisionShot project team. Any

AI-assisted material included in the final submission will be reviewed for accuracy and clearly identified where appropriate.

Appendix C: Additional Diagrams and Materials







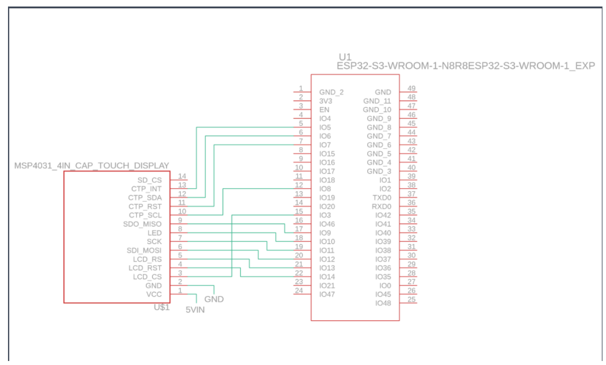


Figure C-1: ESP32-S3-to-ST7796S LCD Interface Design and Pin Connections

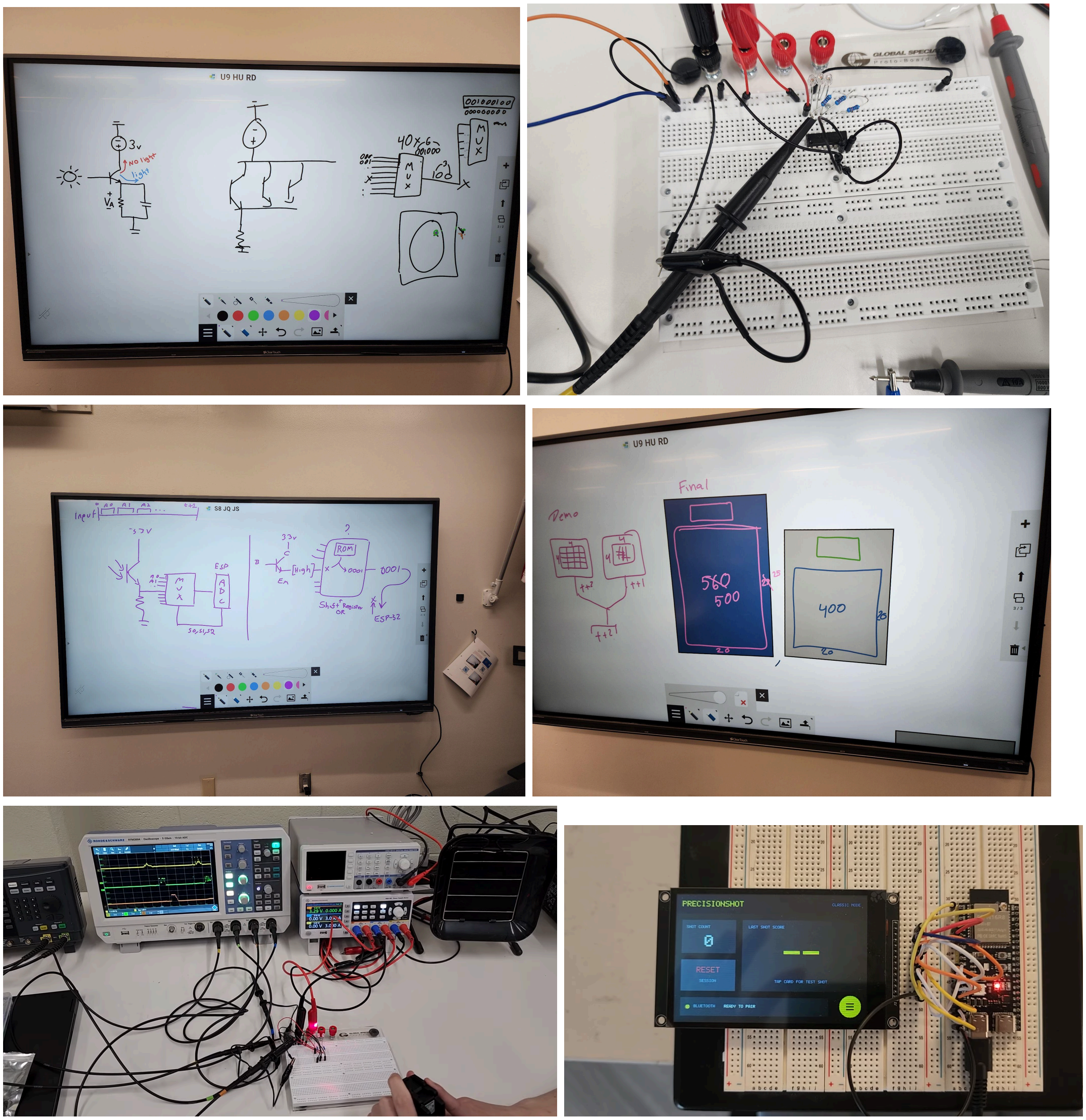


Figure C-2: ESP32-S3-to-ST7796S LCD Interface Design, Testing, and Final Prototype Integration

Appendix D: Project Resources and Repository Links

This appendix provides direct access to the primary online resources associated with the PrecisionShot Training System. These resources include the official project website, public source-code repositories, project-management views, submitted Senior Design documents, and demonstration media. The linked materials provide additional documentation and development history beyond the contents of this report.

Resource	Description	Link
Official Project Website	Main public website containing the project overview, team information, design materials, documents, videos, and timeline	https://www.precisionshot.xyz/
Project Tasks Dashboard	Public task-management page containing assignments, milestones, development progress, and project status	https://www.precisionshot.xyz/tasks
Website Source Repository	Source code for the PrecisionShot public website	https://github.com/Teesitin/precisionshot-web
Mobile Application Repository	React Native, Expo, and TypeScript source code for the companion application	https://github.com/Teesitin/precisionshot-app
Embedded Firmware Repository	ESP32-S3 firmware and embedded-system development files	https://github.com/Teesitin/precisionshot-firmware
Divide and Conquer Document	Initial Senior Design I planning and responsibility document	https://delayne.b-cdn.net/precisionshot/Divide%20%26%20Conquer%20Document%20and%20Committee%20Formation%20-%20Senior%20Design%201%20(12).pdf
Midterm Report	Senior Design I Midterm Milestone Report	https://delayne.b-cdn.net/precisionshot/Midterm%20Report%20-%20Senior%20Design%201%20(1).pdf
Mini Demonstration Video	Senior Design I demonstration of current project progress	https://delayne.b-cdn.net/precisionshot/PrecisionShot_SD1_Mini_Demo_2026-07-27.mp4