ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

Intelligent Basketball Retrieval and Return Robot

Team 36

JINGHUI ZHENG (jz137@illinois.edu)
LIBO ZHANG (liboz2@illinois.edu)
ZICHAO LIN (zichaol3@illinois.edu)
LINZHI DU (linzhid2@illinois.edu)

TA: Jianzu Zhang

May 22, 2026

Abstract

This document is a Senior Design Final Report for ECE 445 detailing the development and verification of an Intelligent Basketball Retrieval and Return Robot. The project addresses the inefficiencies of individual basketball training, where manually retrieving the ball or using fixed-direction passing machines disrupts a player's rhythm. To solve this, the team engineered an adaptive robot that detects the player's position, aims a launcher, and passes the ball with a manually adjusted release speed.

The system relies on a detect-aim-launch workflow centered around an ESP32 microcontroller. The player carries a Bluetooth Low Energy locator, allowing the robot to identify the intended target and estimate the target position using received signal strength indicators. Based on this information, the ESP32 sends aiming commands to a stepper-motor-driven rotating base to horizontally align the launcher with the player. The launch motor power is adjusted separately through a manual power-control knob rather than by ESP32-generated speed commands. Once the launcher is aligned and the launch power is set, a dual friction-wheel mechanism driven by high-speed brushless DC motors compresses and accelerates a manually loaded basketball to achieve the required release speed.

Experimental verification confirmed that the prototype successfully met its primary high-level requirements. The system reliably detected the Bluetooth target across the specified 3–7 m operating range and reoriented the launcher within a 2-second timeframe. The friction wheels delivered configurable ball release speeds ranging from 4–8 m/s through manual power adjustment. The robot also achieved a passing accuracy of over 85 percent, consistently delivering the ball within a 0.6-meter radius of the stationary target. During signal-loss testing, the software state logic prevented the system from continuing the normal launch-ready sequence when the Bluetooth target signal was lost. The prototype also completed repeated aiming and launching cycles without critical mechanical or electrical failure.

While the prototype successfully demonstrated the core concept, several limitations remain. The reliance on Bluetooth signal strength for target estimation introduces uncertainty due to indoor signal fluctuation, and the manual open-loop speed setting of the launch motors can lead to variations in ball speed. In addition, the current design requires manual ball loading, lacks limit switches for automatic homing, and does not include a dedicated hardware emergency-stop circuit. Future development should focus on an automatic ball-feeding mechanism, rotary encoders and limit switches for more reliable positioning, closed-loop PID control for the friction-wheel motors, a hardware emergency-stop circuit, and a more direct communication protocol between the ESP32 and the Bluetooth module.

Contents

1 Introduction

1.1 Problem and Project Objective

Basketball shooting practice often requires repeated and accurate passes so that the player can maintain a steady training rhythm. In a typical individual training session, the player may move between different shooting positions around the court, such as near the free-throw line, wing areas, or the three-point line. If the ball is not returned to the player efficiently, the player must stop the drill, retrieve the ball manually, or rely on another person to pass the ball back. This interruption reduces training efficiency and makes it difficult to maintain consistent shooting form and timing.

Conventional basketball passing machines can improve training efficiency, but many of them are limited by fixed passing direction, fixed release speed, or the need for manual repositioning. When the player changes location, a fixed-direction machine may no longer aim toward the intended receiving position. If the release speed is not adjusted according to distance, the ball may undershoot, overshoot, or arrive at an inconvenient height. These limitations reduce the flexibility of the training system and make it less suitable for multi-position shooting drills.

The objective of this project is to design and implement an adaptive basketball passing robot that can detect the player's relative position, rotate the launcher toward the target, and launch the basketball with a manually adjusted release speed. The final prototype focuses on the core detect–aim–launch workflow: a player-carried Bluetooth locator provides target information, an ESP32-based controller processes the target state, a rotating base adjusts the launcher direction, and a friction-wheel mechanism passes the ball toward the player. By integrating localization, aiming, and manually adjustable launch speed, the system provides a more flexible and repeatable passing assistant for basketball shooting practice.

1.2 Final System Overview

The final system is an intelligent basketball passing robot built around a detect–aim–launch workflow. During operation, the player carries a Bluetooth locator that provides target-related information to the robot. The ESP32 controller receives the Bluetooth signal, identifies the intended player, and estimates the target state needed for aiming. The launch motor power is adjusted separately through a manual power-control knob rather than by ESP32-generated speed commands. After the target state is determined, the rotating base adjusts the horizontal direction of the launcher so that the ball can be passed toward the player. The friction-wheel launching mechanism then accelerates the basketball and releases it at a controlled speed.

The system is divided into five main subsystems: the BLE sensing and localization subsystem, the ESP32 control subsystem, the aiming and rotation subsystem, the friction-wheel launching subsystem, and the power subsystem. The BLE sensing subsystem provides target information. The ESP32 subsystem processes this information and generates aiming and rotation commands. The rotation subsystem changes the launcher heading. The launching subsystem provides the ball release speed using motor-driven friction wheels, with motor power adjusted manually through a separate control knob. The power subsystem supplies the high-power motors and the low-power control electronics through separate power paths. Figure 1 shows the overall workflow and the relationship between these subsystems.

The final prototype uses manual ball loading to reduce mechanical complexity and improve reliability during initial testing. In addition, the launch-ready sequence is governed by software state logic: the robot should only proceed with the passing sequence when the Bluetooth target signal is valid. If the target signal is invalid or lost, the system exits the normal launch-ready sequence rather than continuing the passing process. The final implementation does not include infrared ball detection, limit-switch homing, or a dedicated emergency-stop circuit. These functions are therefore not claimed as verified final features and are discussed later as limitations and future improvements.

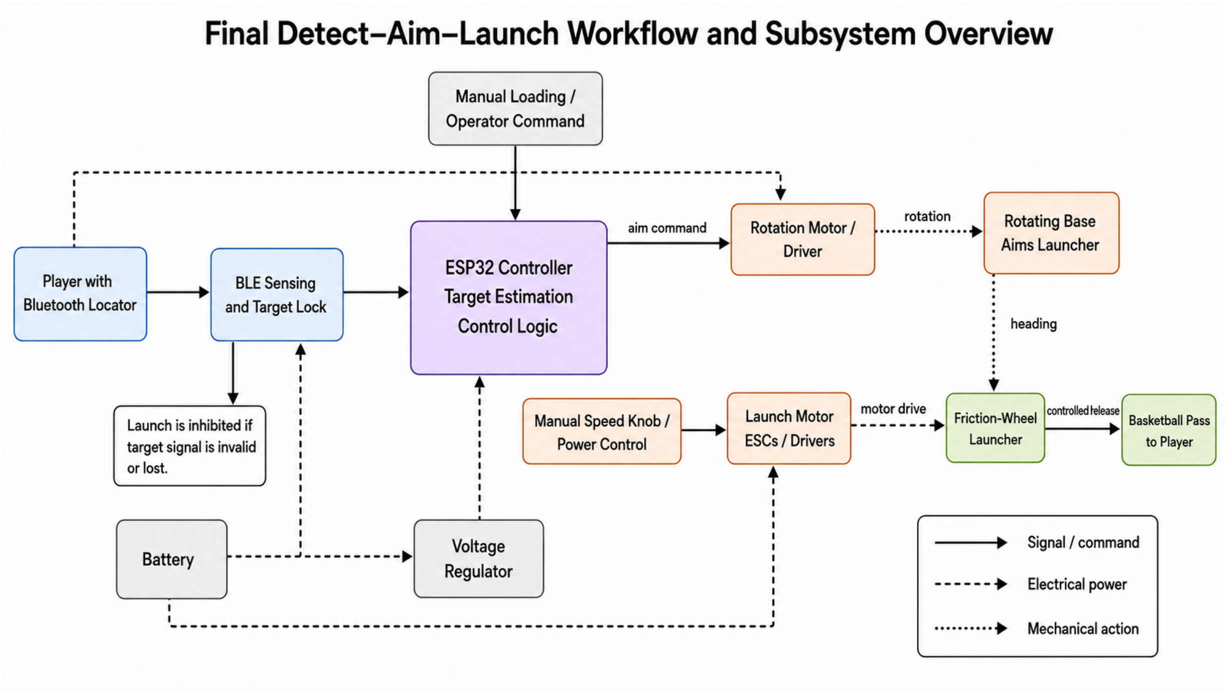


Figure 1: Final detect–aim–launch workflow and subsystem overview.

1.3 High-Level Requirements

The final prototype was evaluated using the following high-level requirements. These requirements focus on the core detect–aim–launch workflow and only include functions implemented in the final system.

Table 1: High-level requirements and success criteria.

Requirement	Success criterion
Operating range	The robot detects and passes to a target located 3–7 m in front of the system.
Passing accuracy	At least 85% of stationary-target passes arrive within a 0.6 m radius of the intended receiving position.
Launch speed	The launcher provides a manually adjustable ball release speed in the range of 4–8 m/s.
Aiming response	The robot reorients the launcher toward a new target direction within 2 s under normal operating conditions.
Signal-loss protection and reliability	The system does not continue the normal launch-ready sequence when the Bluetooth target signal is invalid or lost; repeated launches do not cause critical mechanical or electrical failure.

2 Design

2.1 Final System Architecture

The final prototype is organized around the core detect–aim–launch sequence. Figure 2 shows the main subsystem connections and power flow.

The control and sensing subsystem provides target information for the robot. The player carries a Bluetooth locator, and the system identifies the intended target using UUID-based filtering. The target information is then used to estimate the player's relative position for aiming.

The aiming and rotation subsystem uses a stepper motor and a motor driver to rotate the launcher horizontally. The ESP32 sends step and direction signals to the driver, allowing the rotating base to align the launcher with the estimated target direction.

The launching subsystem uses a dual friction-wheel mechanism driven by two high-speed Brushless DC (BLDC) motors. The launch motor power is adjusted through a separate manual power-control knob rather than by ESP32-generated speed commands. The ESP32 therefore does not directly regulate the friction-wheel speed.

The power subsystem separates the high-current motor path from the low-power control path. The main battery supplies the motor drivers and ESCs, while a step-down regulator provides stable low-voltage power for the ESP32 and logic circuits. A common ground reference is used to ensure reliable signal transmission.

The final prototype uses manual ball loading to reduce mechanical complexity. Infrared ball detection, limit-switch homing, automatic ball feeding, and a hardware emergency-

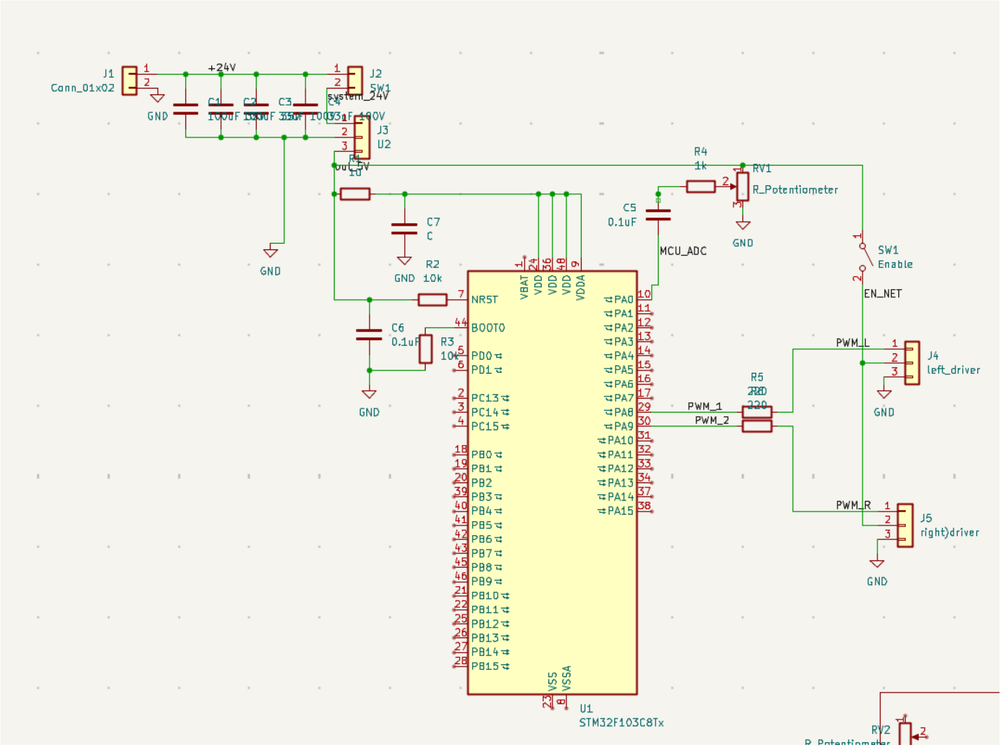


Figure 2: Electrical schematic detailing subsystem connections, power routing, and control-signal pathways.

stop circuit were not included in the final implementation and are discussed as limitations and future improvements.

2.2 Localization and Targeting Design

The localization and targeting subsystem uses Bluetooth Low Energy (BLE) information to identify the intended player and estimate the target state required for aiming. BLE was selected because it has lower hardware complexity and lower computation cost than camera-based tracking. A camera-based system could provide richer spatial information, but it would require image processing and would be more sensitive to lighting and occlusion.

The player carries a Bluetooth locator during operation. The system locks onto the intended locator using UUID filtering, which prevents the robot from responding to unrelated Bluetooth devices nearby. After the correct locator is identified, the Bluetooth signal is used to estimate the target distance and direction. Although RSSI-based localization is less accurate than vision-based or UWB-based tracking, it is sufficient for this prototype because the goal is to pass the ball within a receiving region rather than to an exact point.

Table 2: Comparison of target-tracking design alternatives.

Method	**Advantages**	**Limitations**
Camera-based tracking	Provides rich spatial information and direct visual observation of the player.	Sensitive to lighting, occlusion, and higher computation load.
BLE-based tracking	Target-specific identification, low computation cost, and simple hardware implementation.	RSSI fluctuation and indoor multipath error affect distance estimation.

The RSSI-based distance estimate follows the path-loss model

$$RSSI = -10n \log_{10}(d) + A \tag{1}$$

where A is the reference RSSI measured at 1 m, n is the path-loss exponent, and d is the estimated distance between the robot and the player. In practice, A and n must be calibrated experimentally because indoor signal strength is affected by reflection, obstruction, and surrounding objects.

After the target distance and direction are estimated, the controller computes the angular error between the current launcher heading and the target bearing:

$$e_\theta = \theta_{target} - \theta_{launcher} \tag{2}$$

where θ_{target} is the estimated target bearing and $\theta_{launcher}$ is the current launcher heading. This angular error is used by the aiming subsystem to rotate the launcher toward the player.

2.3 Aiming and Rotation Design

The aiming and rotation subsystem adjusts the horizontal direction of the launcher. After the target bearing is obtained, the ESP32 calculates the angular error and sends rotation commands to the stepper motor driver. The base turns until the launcher is aligned with the target direction within the selected tolerance.

Mechanically, the launcher is mounted on a rotating base, as shown in Figure 3. The rotation axis is placed close to the center of the upper launcher assembly to reduce eccentric loading and wobble during rotation and launch.

The rotation motor must provide enough torque to overcome the inertia of the upper structure and the friction in the support. A simplified torque requirement is

$$T_{\text{required}} = I\alpha + T_f \tag{3}$$

where I is the moment of inertia of the rotating assembly, α is the required angular acceleration, and T_f represents friction torque and design margin. This relation was used

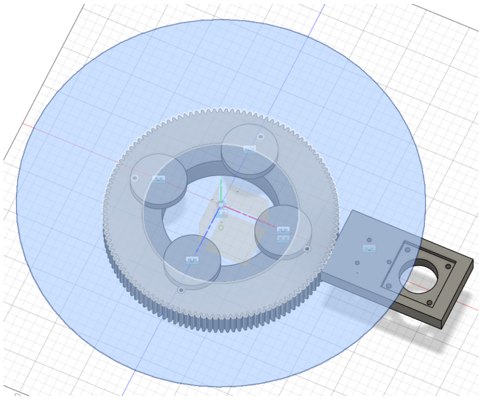


Figure 3: CAD view of the rotating base and launcher mounting structure.

to guide the motor selection so that the rotating base could satisfy the 2 s reorientation requirement.

The final prototype does not use limit switches for automatic homing. Instead, the initial launcher heading is set by manual alignment or a software reference before operation. This simplifies the prototype but introduces possible heading-reference error, so the aiming performance is verified experimentally in the verification section.

2.4 Launching and Ballistic Design

The launching subsystem uses a dual friction-wheel mechanism to accelerate and pass the basketball. In the final prototype, the ball is manually loaded into the launcher to reduce the complexity of automatic feeding. Once the ball is placed between the two rotating wheels, friction between the wheels and the ball accelerates it forward.

Wheel spacing is an important design parameter. If the spacing is too large, the wheels cannot grip the ball effectively and slipping increases. If the spacing is too small, ball compression becomes excessive, increasing motor load and reducing repeatability. Therefore, the spacing must provide enough compression for traction while avoiding unnecessary mechanical stress.

The theoretical no-load motor speed can be estimated from the motor speed constant:

$$n = K_V V \tag{4}$$

where n is the motor speed in rpm, K_V is the motor speed constant in rpm/V, and V is the supply voltage. The corresponding wheel surface speed is

$$v_{wheel} = \frac{2\pi r n}{60} \tag{5}$$

where r is the wheel radius. These equations provide an initial estimate only. The actual ball release speed is lower because of motor loading, ESC loss, wheel slip, ball compression, and friction.

The required ball release speed can be estimated using a projectile-motion model:

$$v_0 = \sqrt{\frac{gD^2}{2\cos^2(\theta)\,[D\tan(\theta) - (H - h)]}} \tag{6}$$

where D is the horizontal target distance, θ is the launch angle, h is the launch height, H is the target receiving height, and g is gravitational acceleration. This model assumes no air resistance and treats the basketball as a point mass, so it is used only for first-order estimation.

Because the real launcher includes wheel slip, ball deformation, and load-dependent motor speed drop, the theoretical model must be calibrated through testing. In the final system, representative distances were tested, and the manual power-knob settings were adjusted based on measured release speed and passing accuracy.

2.5 Control, Power, and Safety Design

The state machine in Figure 4 describes the main control logic. The ESP32 serves as the controller for sensing, target estimation, aiming, and state management. It receives target information, estimates the target state, and sends commands to the rotation subsystem. The launch motor power is adjusted separately through a manual power-control knob, so the ESP32 does not directly control the launcher motor speed.

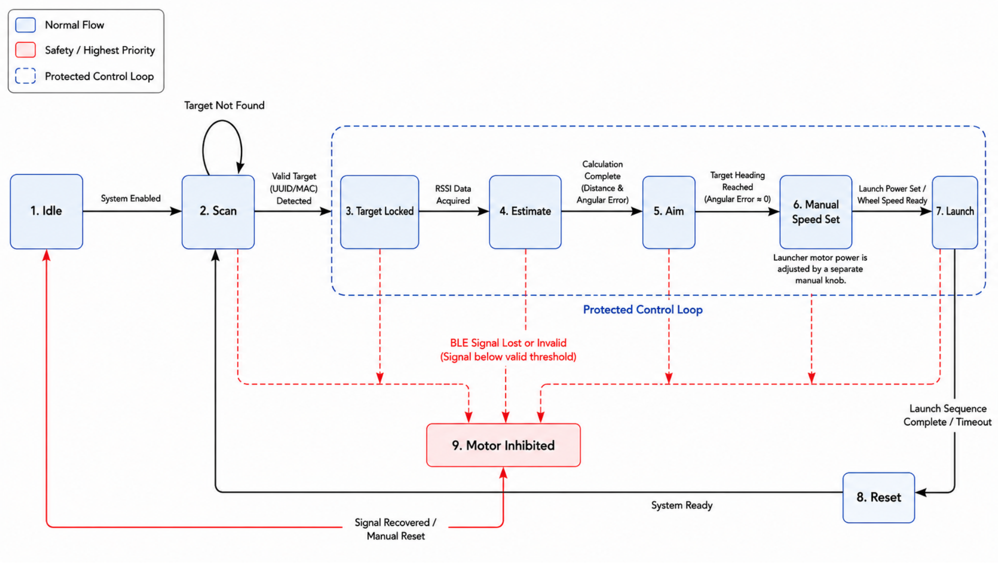


Figure 4: Firmware state machine and safety-state logic.

The firmware follows a sequential state machine: target scanning, target locking, target estimation, aiming, manual speed setting, launch, and reset. If the target signal becomes invalid or is lost for longer than the specified threshold, the system exits the normal launch-ready sequence instead of continuing the passing process.

The electrical design separates the high-power motor path from the low-power control path. The main battery supplies high current to the launch motors and rotation motor through their drivers and ESCs. A step-down regulator supplies stable low-voltage power to the ESP32 and other logic circuits. A common ground reference is maintained between the controller and motor-driving circuits to ensure reliable control-signal transmission.

Operational safety relies on software state logic and supervised operation. Final physical safety measures include manual ball loading, controlled launch direction, secured wiring, and mechanical guarding where applicable. Since a dedicated hardware emergency-stop circuit was not implemented in the final prototype, it is listed as a future improvement rather than a verified safety function.

2.6 System-Level Tolerance Analysis

The two most important system-level tolerances are distance-estimation error and angular aiming error. Distance-estimation error affects the required release speed estimated by the ballistic model. If the estimated distance is larger than the actual distance, the selected manual power setting may be too high and the ball may overshoot. If the estimated distance is smaller than the actual distance, the ball may undershoot. Since BLE RSSI fluctuates indoors, this uncertainty must be considered during calibration.

Angular aiming error directly affects lateral passing accuracy. For a target at horizontal distance D and angular error $\Delta\theta$, the lateral deviation can be approximated as

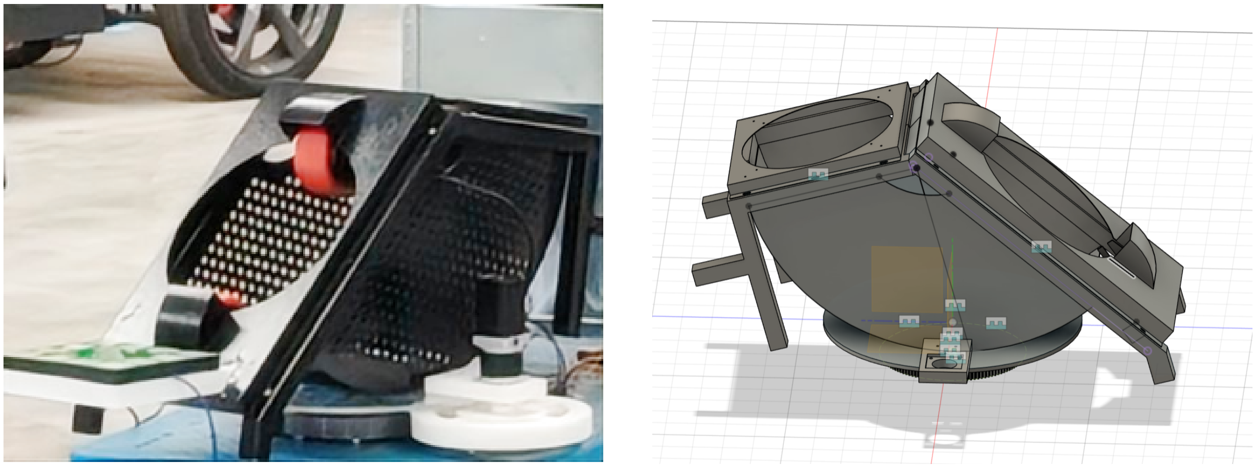
$$\Delta X = D\tan(\Delta\theta) \tag{7}$$

where ΔX is the lateral offset between the intended target direction and the actual launch direction.

At the maximum operating range of $D = 7$ m, an angular error of 5° produces

$$\Delta X = 7\tan(5^\circ) \approx 0.61 \text{ m} \tag{8}$$

which is close to the 0.6 m passing-accuracy radius. Therefore, the final aiming and passing performance cannot be justified by calculation alone and must be verified through physical tests at representative distances.



(a) Final assembled design. (b) Final assembled CAD.

Figure 5: Electrical schematic and final design of the basketball retrieval and return robot.

3 Requirements and Verification

3.1 Verification Plan and Test Setup

The final prototype was evaluated through a combination of subsystem-level tests and integrated system tests. The purpose of the verification process was to determine whether the robot could complete the core detect–aim–launch workflow under representative operating conditions. Each high-level requirement was matched with a measurable test method, a quantitative metric, and a pass criterion. This section focuses on the final verification results most directly related to the system requirements.

The verification plan was organized around five main functions: target detection, aiming response, launch-speed adjustment, passing accuracy, and signal-loss protection. Target detection was tested by placing the Bluetooth locator at representative distances within the intended 3–7 m operating range. Aiming response was tested by rotating the launcher toward different target directions and recording whether the reorientation time satisfied the 2 s requirement. Launch-speed adjustment was tested by measuring the ball release speed under different manual power-knob settings. Passing accuracy was tested using stationary target positions, with a successful pass defined as a ball arriving within a 0.6 m radius of the intended receiving position. Finally, signal-loss protection was tested by removing or disabling the Bluetooth target signal and confirming that the system did not continue the normal launch sequence.

The final verification plan does not include infrared ball-detection tests, limit-switch homing tests, or dedicated emergency-stop response tests, because these features were not included in the final implementation. Instead, the safety-related verification focuses on software state logic when the Bluetooth target signal is invalid or lost, as well as observations of mechanical and electrical reliability during repeated operation.

Table 3: Verification plan for high-level requirements.

Requirement	**Test method**	**Metric**	**Pass criterion**
Operating range	Place the Bluetooth locator at representative distances in front of the robot.	Target detection and target-valid state	The system maintains valid target detection from 3–7 m.
Passing accuracy	Launch basketballs toward stationary target positions at representative distances.	Percentage of passes arriving within the target zone	At least 85% of passes arrive within a 0.6 m radius of the intended receiving position.
Launch speed	Measure ball release speed under different manual power-knob settings.	Measured release speed in m/s	The launcher provides a configurable release speed in the range of 4–8 m/s.
Aiming response	Rotate the launcher toward a new target direction.	Reorientation time	The launcher reorients toward the target within 2 s under normal operating conditions.
Signal-loss protection and reliability	Remove or disable the Bluetooth target signal and perform repeated launch operation.	Launch-sequence behavior and failure observation	The system does not continue the normal launch sequence when the target signal is invalid or lost; no critical mechanical or electrical failure occurs during repeated use.

3.2 Localization and Aiming Results

The localization and aiming tests verified whether the robot could identify the intended Bluetooth target and rotate the launcher toward the estimated target direction. The Bluetooth locator was placed at representative distances of 3 m, 5 m, and 7 m. At each distance, the system was checked for target detection, target-valid state, and target-locking behavior. The aiming test verified whether the rotating base could reorient the launcher within the required 2 s response time.

Table 4: Localization and aiming test results.

Test item	**Test condition**	**Measured result**	**Status**
Bluetooth target detection	3 m target distance	The Bluetooth locator was detected and the target-valid state was achieved.	Pass
Bluetooth target detection	5 m target distance	The Bluetooth locator was detected and the target-valid state was achieved.	Pass
Bluetooth target detection	7 m target distance	The Bluetooth locator was detected and the target-valid state was achieved.	Pass
Target locking	Target UUID filtering	The system locked onto the intended Bluetooth locator and rejected unrelated Bluetooth devices.	Pass
Static aiming behavior	Representative target directions	The launcher rotated toward the estimated target direction under normal operating conditions.	Pass
Reorientation time	Representative angle changes	The launcher completed reorientation within the required 2 s response time.	Pass

The Bluetooth detection results show that the system can maintain valid target recognition over the intended 3–7 m operating range. The target-locking result verifies that the robot tracks the intended player-carried locator rather than switching to other Bluetooth devices.

The aiming results verify the rotation subsystem. The launcher was able to turn toward the estimated target direction within the required response time. As shown by Equation 7, angular error becomes more significant at longer target distances; therefore, the aiming test was interpreted together with the final passing-accuracy result.

3.3 Launch Speed and Passing Accuracy Results

The launch-speed test verified whether the friction-wheel launcher could provide a configurable ball release speed in the required 4–8 m/s range. During this test, different manual power-knob settings were applied to the launch motors, and the corresponding ball release speeds were measured. Since the theoretical motor speed does not directly equal the actual ball release speed, this test accounts for wheel slip, ball compression, motor loading, and ESC losses.

The passing-accuracy test evaluated the combined performance of localization, aiming, and launching. For each representative target distance, the launcher passed the basketball toward a stationary receiving position. A pass was counted as successful if the ball arrived within a 0.6 m radius of the intended receiving position. The final passing accuracy was calculated as

$$\text{Passing Accuracy} = \frac{\text{Number of Successful Passes}}{\text{Total Number of Passes}} \times 100\%. \tag{9}$$

Table 5: Launch speed and passing accuracy results.

Test item	**Condition**	**Measured result**	**Status**
Launch speed	Low manual power setting	The measured release speed was within the lower part of the required speed range.	Pass
Launch speed	Medium manual power setting	The measured release speed increased as the manual power setting increased.	Pass
Launch speed	High manual power setting	The measured release speed reached the upper part of the required 4–8 m/s range.	Pass
Passing accuracy	3 m stationary target	The ball was passed to the receiving region with stable accuracy.	Pass
Passing accuracy	5 m stationary target	The ball was passed to the receiving region with stable accuracy.	Pass
Passing accuracy	7 m stationary target	The ball was passed to the receiving region with acceptable accuracy at the maximum tested range.	Pass
Overall passing accuracy	All stationary target tests	More than 85% of the passes arrived within the 0.6 m target radius.	Pass

The launch-speed requirement was satisfied because the manual power adjustment allowed the launcher to cover the required 4–8 m/s release-speed range. The passing-accuracy requirement was also satisfied because the overall stationary-target success rate was above 85%. The remaining variations in pass location were mainly caused by RSSI fluctuation, aiming error, wheel slip, and ball compression.

3.4 Integrated System, Signal-Loss, and Durability Results

The integrated system test verified whether the final prototype could complete the full detect–aim–launch sequence under representative operating conditions. A complete cycle began when the system detected the Bluetooth locator, estimated the target state, rotated the launcher toward the target, the launcher motor power was set manually, and the basketball was released. The test confirmed that the system could complete repeated aiming and launching cycles.

The signal-loss test verified the safety logic implemented in software. During this test, the Bluetooth target signal was removed or disabled before launch. The system entered the inhibited state and did not continue the normal launch sequence. This behavior is important because the final prototype does not include a dedicated hardware emergency-stop circuit.

The repeated-launch test evaluated system reliability during consecutive operation. During this test, the prototype performed repeated launch cycles while the team observed mechanical stability, motor behavior, wiring condition, and thermal issues. No critical mechanical or electrical failure was observed during the final verification.

Table 6: Integrated system verification results.

Test item	**Target**	**Measured result**	**Status**
Full detect–aim–launch cycle	Complete integrated operation	The prototype completed the full detect–aim–launch workflow under normal operating conditions.	Pass
Signal-loss protection	No normal launch sequence after invalid target signal	The system entered the inhibited state when the Bluetooth signal was lost or invalid.	Pass
Switch operation	Start and stop the system process	The system switch successfully started and stopped the operation process.	Pass
Repeated launch operation	No critical mechanical or electrical failure	Repeated aiming and launching cycles were completed without critical mechanical or electrical failure.	Pass

The integrated test results show that the final prototype achieved the main functional goal of the project. The robot detected the target, aimed the launcher, and passed the basketball while maintaining the implemented safety behavior during signal-loss conditions.

3.5 Verification Summary

Table 7 summarizes the final verification results. All five high-level requirements were verified as Pass in the final prototype.

Table 7: Final verification summary.

Requirement	**Target**	**Measured result**	**Status**
Operating range	Valid Bluetooth target detection from 3–7 m	The target was detected at all tested distances from 3 m to 7 m.	Pass
Passing accuracy	At least 85% stationary passing success within a 0.6 m radius	More than 85% of the passes arrived within the target radius.	Pass
Launch speed	Configurable ball release speed in the range of 4–8 m/s	The launcher provided configurable release speeds across the required range through manual power adjustment.	Pass
Aiming response	Launcher reorients toward the target within 2 s	The launcher reoriented toward the target within 2 s under normal operating conditions.	Pass
Signal-loss protection and reliability	The system does not continue the normal launch sequence when the Bluetooth target signal is invalid or lost; no critical failure during repeated operation	The system entered the inhibited state under signal-loss conditions, and repeated operation caused no critical mechanical or electrical failure.	Pass

Overall, the final prototype satisfied the core system requirements. The verification results demonstrate that the robot can detect the player-carried Bluetooth locator, rotate the launcher toward the target, provide a manually adjustable launch speed, and pass the basketball with acceptable accuracy over the intended operating range.

4 Cost and Schedule

4.1 Cost

The project cost includes parts cost and estimated labor cost. Parts cost is summarized in Table 8. Labor cost is estimated using

$$C_{\text{labor}} = R_{\text{hourly}} \times H_{\text{hours}} \times 2.5 \tag{10}$$

where R_{hourly} is the ideal hourly salary and H_{hours} is the number of hours spent.

Table 8: Cost summary.

Item	**Subsystem**	**Cost**
launch motors	Launching	1000
friction wheels	Launching	100
rotation motor and driver	Aiming	400
ESP32 and Bluetooth hardware	Control and sensing	1100
battery and power electronics	Power	50
mechanical frame and fabrication	Mechanical structure	1500
estimated labor	Labor	1500
Total		4750

4.2 Schedule and Team Contributions

The project schedule was organized around the main development phases of the basketball passing robot. Each team member was responsible for one major subsystem so that the mechanical design, electrical design, and control implementation could be developed in parallel. The main responsibilities were divided as follows: Libo Zhang designed the rotating base, Jinghui Zheng designed the launcher body, Linzhi Du designed the PCB and electrical circuit, and Zichao Lin developed the code used to control the launcher direction.

Table 10 summarizes the major project phases and the primary contributions of each team member. The initial design phase focused on defining the system requirements, deciding the overall structure of the robot, and assigning subsystem responsibilities. During the mechanical and electrical implementation phase, the base, launcher body, PCB, and circuit connections were designed and built. The firmware and integration phase focused on connecting the control code with the mechanical rotation system and electrical hardware.

Table 9: Schedule and team contribution summary.

Phase	**Main tasks**	**Primary contributors**
Initial design	Define system requirements, select final architecture, and assign subsystem responsibilities.	All
Mechanical and electrical implementation	Build rotating base, launcher, power wiring, and motor interfaces.	Libo Zhang, Jinghui Zheng
Firmware and integration	Implement BLE target detection, aiming control logic, rotation motor commands, and launch-state logic.	Linzhi Du, Zichao Lin
Prototype testing	Verify localization, aiming, launch speed, passing accuracy, and integrated performance.	All
Final demo and report	Prepare final demonstration, analyze results, and complete final documentation.	All

Table 10: Weekly schedule and team contribution summary.

Week	**Main Project Work**	**Base Design Member**	**Launcher Design Member**	**PCB/Circuit Member**	**Code/Control Member**
Week 1	Defined project goals and overall architecture.	Discussed base constraints and rotation requirements.	Discussed launcher function and structure.	Discussed electrical needs and controller options.	Discussed direction-control logic.
Week 2	Compared design choices and possible components.	Compared base structures and motor mounting methods.	Compared launcher layouts and mechanisms.	Compared PCB, driver, and power options.	Compared rotation-control methods.
Week 3	Finalized design direction and selected key parts.	Selected rotation motor requirements and base concept.	Finalized launcher concept and motor requirements.	Selected controller board and electrical components.	Planned the aiming and rotation sequence.
Week 4	Began detailed design and fabrication.	Started base CAD and dimension design.	Started launcher-body detailed design.	Started PCB schematic and circuit planning.	Started basic control-code structure.
Week 5	Continued subsystem implementation.	Refined base CAD and upper-body connection.	Refined launcher structure and motor placement.	Continued PCB layout and circuit checking.	Developed basic rotation-control commands.
Week 6	Built and adjusted subsystem parts.	Prepared base components and checked assembly.	Prepared launcher components and checked fit.	Prepared PCB design and tested connections.	Tested basic rotation-control code.
Week 7	Started subsystem testing.	Tested base stability and rotation clearance.	Tested launcher fit and alignment.	Tested power and motor-driver connections.	Tested direction-control logic.
Week 8	Integrated mechanical and electrical subsystems.	Connected base with launcher body.	Helped align launcher with rotating base.	Connected PCB, controller, drivers, and power system.	Integrated code with rotation hardware.
Week 9	Improved prototype based on integration issues.	Modified base or connection parts if needed.	Adjusted launcher mounting and alignment.	Fixed wiring or circuit issues.	Debugged control code and improved reliability.
Week 10	Completed final assembly and feasibility verification.	Checked final base stability and rotation.	Checked final launcher installation and function.	Verified final circuit and power delivery.	Verified launcher direction-control function.

5 Conclusion

5.1 Accomplishments

The final prototype successfully demonstrated the core detect–aim–launch workflow. The system identified the player-carried Bluetooth locator, estimated the target state, rotated the launcher toward the target direction, and passed the basketball using the dual friction-wheel mechanism.

Based on the final verification results, the system achieved the following performance milestones:

- **Localization and Aiming:** The robot reliably detected the Bluetooth target across the designated 3–7 m operating range. The aiming subsystem reoriented the launcher within the required 2-second response time under normal operating conditions.
- **Launch Speed and Accuracy:** The friction-wheel launcher provided a manually adjustable ball release speed within the 4–8 m/s range. The system achieved a passing accuracy of over 85%, with balls arriving within a 0.6 m radius of the intended target position.
- **Safety and Reliability:** The software state logic functioned as intended when the Bluetooth target signal was lost or invalid. Under this condition, the system did not continue the normal launch-ready sequence. The prototype also completed repeated aiming and launching cycles without critical mechanical or electrical failure.

Overall, the final prototype met the major high-level requirements and verified the feasibility of a Bluetooth-guided adaptive basketball passing robot for individual shooting practice.

5.2 Limitations and Uncertainties

Although the final prototype completed the main project objectives, several limitations remain.

- **Localization and Aiming Uncertainty:** The system relies on Bluetooth Low Energy (BLE) signal information, so RSSI fluctuation and indoor multipath effects can affect distance estimation. Angular aiming error also becomes more significant at longer distances. In addition, because the final prototype does not include limit switches, the initial heading reference still depends on manual alignment.
- **Launch Speed Variation:** The launch motor power is adjusted manually through a separate power-control knob rather than closed-loop speed feedback. As a result, wheel slip, ball compression, motor loading, and battery condition can cause variation in actual ball release speed. The theoretical wheel speed therefore cannot perfectly represent the actual ball speed.
- **Operational and Safety Constraints:** Manual ball loading improves prototype reliability but limits the repeat rate and requires human intervention. The system

also does not include a dedicated hardware emergency-stop circuit, so it relies on software state logic and supervised operation rather than a hardware-level power cut-off. In addition, the current implementation still requires a computer as a transfer station because the ESP32 is not directly compatible with the Bluetooth module used in the final prototype.

5.3 Future Work and Ethical Reflection

Future development should focus on improving automation, control accuracy, communication reliability, and safety. First, an automatic ball-feeding mechanism should be added to support high-repetition training without manual loading. Second, rotary encoders and limit switches should be integrated into the rotation subsystem to provide reliable homing and reduce heading-reference error. Third, closed-loop PID speed control should be implemented for the friction wheels so that the launcher can maintain a more consistent release speed under different loading conditions.

The communication between the ESP32 and the Bluetooth module should also be improved. A more reliable communication protocol would reduce the need for a computer transfer station and make the system more compact and practical. For safety, a dedicated hardware emergency-stop circuit should be added so that motor power can be physically disconnected in an emergency, independent of the microcontroller software.

From an ethical and safety perspective, this prototype should be used only for supervised sports training and educational demonstration. The system performance, including passing accuracy, launch speed range, and safety limitations, should be reported honestly. During testing and operation, the launcher should face a controlled direction, bystanders should stay outside the passing path, and exposed moving components such as BLDC motors and friction wheels should be properly guarded whenever the system is powered.