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 15, 2026

# Abstract

This document is a Senior Design Final Report for ECE 445 detailing the development 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 dynamically detects the player's position, aims a launcher, and passes the ball with a controlled 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 their distance and bearing using received signal strength indicators. Based on this data, a stepper motor rotates the base to horizontally align the launcher with the player. Once aimed, a dual friction-wheel mechanism driven by high-speed brushless DC motors compresses and accelerates a manually loaded basketball to achieve the necessary release speed.

Experimental verification confirmed that the prototype successfully met its primary high-level requirements. The system reliably detected the Bluetooth target across a specified 3 to 7 meter operating range and reoriented the launcher within a 2-second timeframe. The friction wheels delivered configurable ball release speeds ranging from 4 to 8 meters per second. Consequently, the robot achieved a passing accuracy of over 85 percent, consistently delivering the ball within a 0.6 meter radius of the stationary target. The system also proved reliable during repeated operations and successfully utilized software-based safety interlocks to inhibit launches whenever the target signal was lost.

While the prototype successfully demonstrated the core concept, the report acknowledges several mechanical and control limitations. The reliance on Bluetooth signal strength for distance estimation introduces uncertainties due to indoor signal fluctuations, and the open-loop control of the launch motors can lead to variations in ball speed. Additionally, the current design requires manual ball loading and lacks a dedicated hardware emergency-stop circuit. Future development phases propose replacing the Bluetooth tracking with Ultra-Wideband or computer vision sensors, implementing closed-loop PID control for the motors, adding a physical emergency stop, and building an automated ball-feeding mechanism to maximize training efficiency.

# 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 controlled 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 launch-speed control, the system aims to provide 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 and launch-speed selection. 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 control commands. The rotation subsystem changes the launcher heading. The launching subsystem controls the ball release speed using motor-driven friction wheels. 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 sequence is controlled by software logic: the robot should only allow launch when the Bluetooth target signal is valid. If the target signal is invalid or lost, the system inhibits launch rather than continuing the passing sequence. 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.

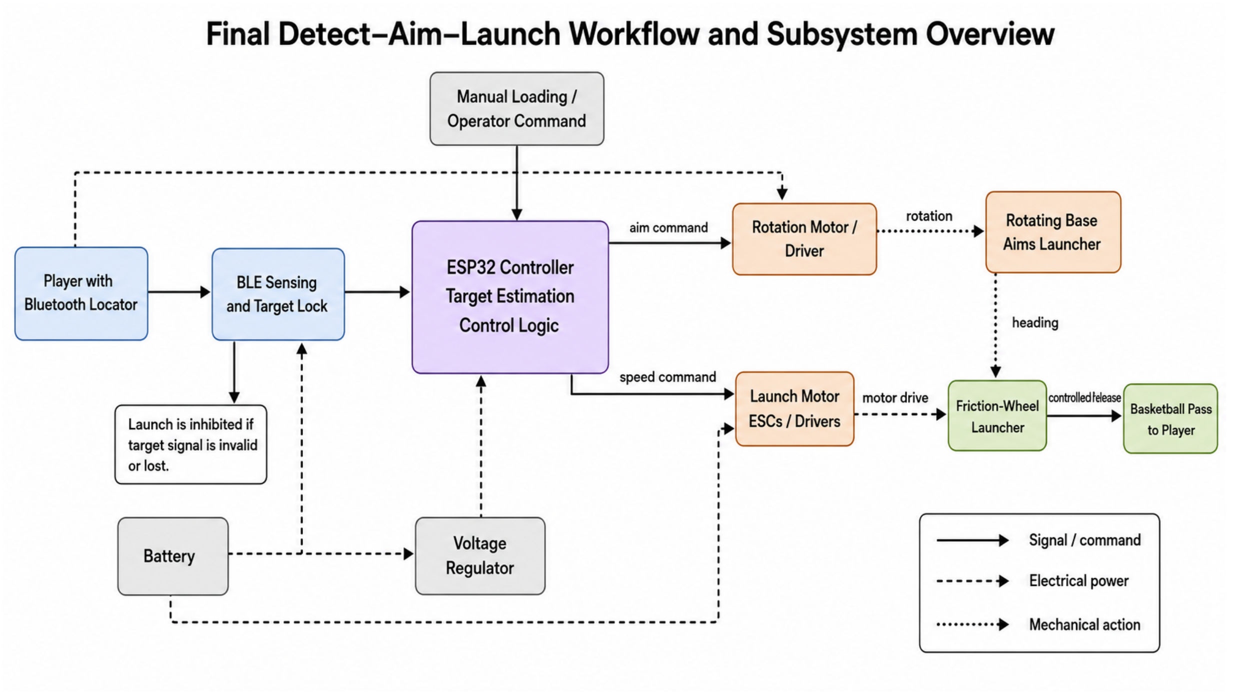

Figure 1: Final detect–aim–launch workflow and subsystem overview.

## 1.3 High-Level Requirements

The final prototype is 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 configurable 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 | Launch is inhibited 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. The Figure2 below illustrate the system.

The Control and Sensing Subsystem acts as the primary data processing hub. The ESP32 utilizes its integrated Bluetooth Low Energy (BLE) module to scan for and lock onto a specific UUID broadcasted by a player-carried locator. Upon receiving the signal, the ESP32 calculates the target's relative distance and bearing. This target state is then translated into actionable kinematic commands for the actuators.

To horizontally align the launcher with the detected target, the Aiming and Rotation Subsystem employs a stepper motor paired with a dedicated motor driver. The ESP32 sends precisely timed Pulse Width Modulation (PWM) step and direction signals to the driver. This stepping mechanism allows for high-resolution angular adjustments, enabling the rotating base to quickly orient the launcher toward the player without relying on complex closed-loop feedback.

The Launching Subsystem governs the ballistic trajectory and release speed of the basketball. It features a dual friction-wheel mechanism driven by two high-speed Brushless DC (BLDC) motors. These motors are regulated by Electronic Speed Controllers (ESCs). The ESP32 dispatches independent PWM signals to the ESCs, adjusting the duty cycle to modulate the wheels' rotational velocity (RPM) based on the calculated target distance. The counter-rotating wheels compress and accelerate the manually loaded basketball to achieve the desired release speed.

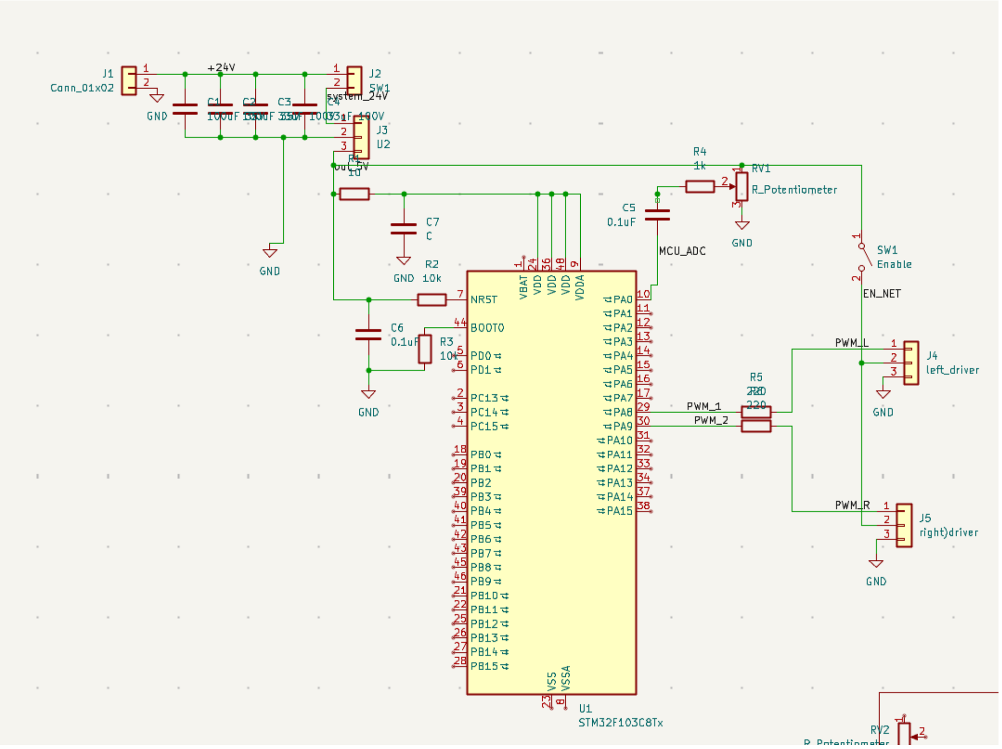

Figure 2: Electrical schematic detailing subsystem connections and PWM signal pathways.

The Power Distribution Network is engineered to isolate sensitive logic components from high-current switching noise. A main high-capacity battery supplies high-current power directly to the ESCs and the stepper motor driver. Simultaneously, a step-down voltage regulator steps this voltage down to a stable low-voltage level (5V/3.3V) to power the ESP32 and peripheral logic circuits

The final prototype uses manual loading to reduce feeding-mechanism complexity and uses target-valid software logic to inhibit launch when the Bluetooth signal is invalid. Infrared ball detection, limit-switch homing, and a dedicated emergency-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 and launch-speed selection. BLE was selected instead of camera-based tracking because it has lower hardware complexity, lower computation cost, and direct compatibility with the ESP32 controller. A camera-based system could provide richer spatial information, but it would also require more image processing, be more sensitive to lighting conditions, and be affected by player occlusion.

The player carries a Bluetooth locator during operation. The ESP32 scans for nearby BLE devices and locks onto the intended locator using UUID or MAC address filtering. This prevents the robot from responding to unrelated Bluetooth devices in the environment. After the correct locator is identified, the received signal strength indicator (RSSI) is used as a rough estimate of the target distance. Although RSSI-based localization is not as accurate as vision-based or UWB-based tracking, it is sufficient for the prototype because the final system only needs to pass the ball within a reasonable 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 direct compatibility with ESP32. | 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, the values of $A$ and $n$ must be calibrated experimentally because indoor signal strength can vary due to multipath reflection, body 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 $\theta_{target}$ 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 before the launch command is enabled.

## 2.3 Aiming and Rotation Design

The aiming and rotation subsystem adjusts the horizontal direction of the launcher so that the ball can be passed toward the detected player. After the BLE subsystem estimates the target bearing, the ESP32 calculates the angular error between the launcher heading and the target direction. The rotating base then turns the launcher until the error is reduced to an acceptable range before launch is enabled.

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 limit wobble during aiming and ball release. This helps improve alignment stability and reduces the effect of structural vibration on passing accuracy.

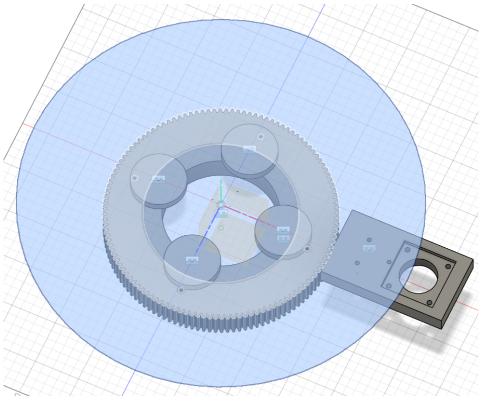

Figure 3: CAD view of the rotating base and launcher mounting structure.

The rotation motor must provide enough torque to overcome the inertia of the upper structure and the friction in the mechanical 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, $\alpha$ is the required angular acceleration, and $T_f$ represents friction torque and design margin. This relation was used to guide the motor selection so that the rotating base could satisfy the 2 s reorientation requirement under normal operating conditions.

The rotation control logic uses the angular error defined in Equation **??**. When a valid target is detected, the ESP32 sends the rotation command to the motor driver. Once the launcher reaches the target direction within the selected tolerance, the system enters the ready-to-launch state. If the target signal becomes invalid during aiming, the firmware exits the launch sequence and inhibits firing.

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 reduces hardware complexity but introduces heading-reference uncertainty, so the final aiming performance is verified experimentally in Section 3.2.

## 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 and improve reliability during early-stage testing.

Once the ball is placed between the two rotating wheels, friction between the wheels and the basketball accelerates the ball forward.

The spacing between the two wheels is an important mechanical design parameter. If the wheel spacing is too large, the wheels cannot provide enough normal force and the ball may slip during launch. If the spacing is too small, the ball compression becomes excessive, which increases motor load and may reduce repeatability. Therefore, the wheel spacing must provide enough compression for traction while avoiding unnecessary mechanical stress on the motors and frame.

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 approximated by

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

where $r$ is the wheel radius. These equations provide an initial estimate of the available wheel speed, but the actual ball release speed is lower than the theoretical no-load value 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, $\theta$ 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 mainly used for first-order launch-speed estimation.

Because the real launcher includes wheel slip, ball deformation, and load-dependent motor speed drop, the theoretical model must be calibrated through experimental testing. In the final system, representative target distances are tested and the corresponding motor commands are adjusted based on measured launch speed and passing accuracy.

## 2.5 Control, Power, and Safety Design

The state machine diagram below indicates how the system keeps itself safe while operating under control. The ESP32 serves as the central controller. It receives BLE target information, estimates the target state, sends commands to the rotation subsystem, and commands the launch motors according to the required release speed. The firmware follows a sequential state machine: target scanning, target locking, target estimation, aiming, launch-speed command, launch, and reset. If the target signal becomes invalid or is lost for longer than the specified threshold, the launch command is inhibited.

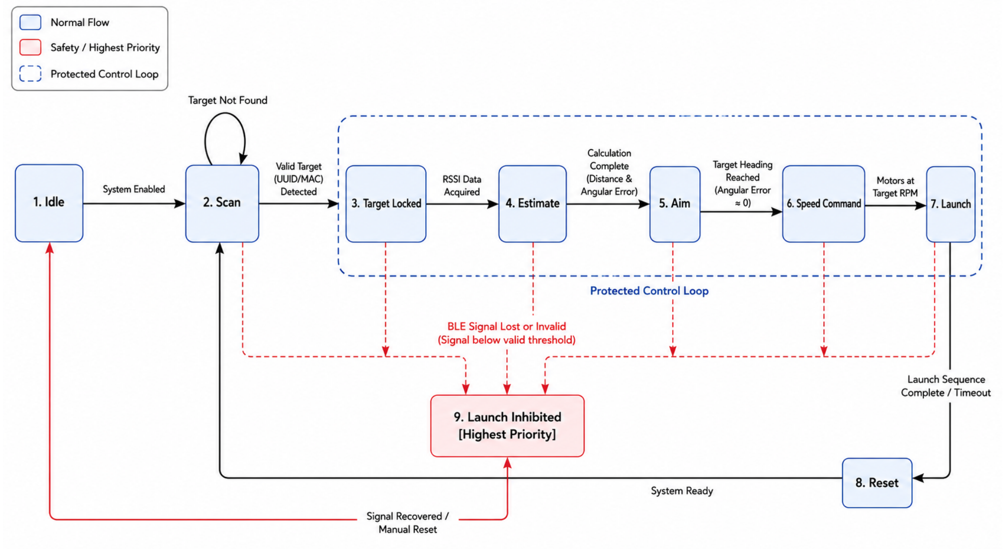

Figure 4: Firmware state machine and launch-inhibition logic.

To ensure stability, the electrical design strictly separates the high-power motor path from the low-power control path. The main battery supplies high current directly to the launch motors and rotation motor through their respective drivers and ESCs. Concurrently, a step-down voltage regulator converts the main voltage to supply stable power to the ESP32 and other low-power electronics. A common ground reference is maintained between the controller and the motor-driving circuits to guarantee clean PWM signal transmission and mitigate electrical noise from the high-power actuators.

Operational safety relies on a combination of software interlocks and physical procedures. As a primary software fail-safe, the launch command is strictly inhibited if the Bluetooth target signal becomes invalid or is lost for longer than the specified threshold. Final physical safety measures include manual ball loading, operating the launcher in a controlled test direction, securing all wiring, and utilizing mechanical guarding where applicable. Because a dedicated hardware emergency-stop circuit was not implemented in the final prototype, it is not claimed as a verified safety function, making the software launch inhibition and supervised testing environments essential for safe operation.

## 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 calculated by the ballistic model. If the estimated distance is larger than the actual distance, the selected launch speed may be too high and the ball may overshoot the target. If the estimated distance is smaller than the actual distance, the ball may undershoot the target. Since BLE RSSI can fluctuate indoors, this distance uncertainty must be considered during experimental 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 $\Delta X$ is the lateral offset between the intended target direction and the actual launch direction.

This relationship shows that angular error becomes more significant as target distance increases. For example, at the maximum operating range of $D = 7$ m, an angular error of $5^\circ$ 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. This analysis motivates the stationary-target passing tests described in the verification section.

# 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. Detailed raw data, extended trial logs, and additional low-level measurements can be included in the appendix, while 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 control, 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 commanding the launcher to rotate toward different target directions and recording the reorientation time and final alignment error. Launch-speed control was tested by measuring the ball release speed under different motor command levels. 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 inhibited launch.

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 launch inhibition 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 rate 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 launch motor command levels. | Measured release speed in m/s | The launcher provides a configurable release speed in the range of 4–8 m/s. |
| Aiming response | Command the launcher to rotate toward a new target direction. | Reorientation time and final alignment error | 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-inhibition behavior and failure observation | Launch is inhibited 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 verify whether the robot can identify the intended Bluetooth target and rotate the launcher toward the estimated target direction. The Bluetooth locator will be placed at representative distances of 3 m, 5 m, and 7 m. At each distance, the system will record target detection, target-valid state, and target-locking behavior. The aiming test will measure the reorientation time and final angular error for representative target directions.

Table 4: Localization and aiming test results.

| **Test item** | **Test condition** | **Measured result** | **Status** |
|---|---|---|---|
| Bluetooth target detection | 3 m target distance | TODO: insert detection rate or target-valid result | TODO |

Table 4: Localization and aiming test results (continued).

| **Test item** | **Test condition** | **Measured result** | **Status** |
|---|---|---|---|
| Bluetooth target detection | 5 m target distance | TODO: insert detection rate or target-valid result | TODO |
| Bluetooth target detection | 7 m target distance | TODO: insert detection rate or target-valid result | TODO |
| Target locking | Target UUID or MAC filtering | TODO: insert whether the system locked only onto the intended locator | TODO |
| Static aiming accuracy | Representative target angles | TODO: insert mean and maximum angular error | TODO |
| Reorientation time | Largest tested angle change | TODO: insert average and maximum reorientation time | TODO |

The Bluetooth detection results will determine whether 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 evaluate the rotation subsystem. The measured reorientation time should be compared with the 2 s requirement, and the final angular error should be interpreted together with the passing-accuracy requirement. As shown by Equation **??**, angular error becomes more significant at longer target distances; for example, a 5 degree aiming error at 7 m produces about 0.61 m of lateral deviation, which is close to the 0.6 m success-radius requirement.

After testing is completed, the final status of this subsection should be reported as Pass, Partial, or Fail based on the measured detection rate, target-locking behavior, reorientation time, and angular error in Table 4.

## 3.3 Launch Speed and Passing Accuracy Results

The launch-speed test verifies whether the friction-wheel launcher can provide a configurable ball release speed in the required 4–8 m/s range. During this test, different motor command levels are applied to the launch motors, and the corresponding ball release speeds are measured. The measured values are then compared with the required speed range. Since the theoretical motor speed does not directly equal the actual ball release speed, this test is necessary to account for wheel slip, ball compression, motor loading, and ESC losses.

The passing-accuracy test evaluates the combined performance of localization, aiming, and launching. For each representative target distance, the launcher is commanded to pass the basketball toward a stationary receiving position. A pass is counted as successful if the ball arrives within a 0.6 m radius of the intended receiving position. The final passing accuracy is 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 command level | TODO: insert measured release speed | TODO |
| Launch speed | Medium command level | TODO: insert measured release speed | TODO |
| Launch speed | High command level | TODO: insert measured release speed | TODO |
| Passing accuracy | 3 m stationary target | TODO: insert successful passes / total trials | TODO |
| Passing accuracy | 5 m stationary target | TODO: insert successful passes / total trials | TODO |
| Passing accuracy | 7 m stationary target | TODO: insert successful passes / total trials | TODO |
| Overall passing accuracy | All stationary target tests | TODO: insert total success rate | TODO |

The launch-speed requirement is satisfied if the measured release-speed range covers 4–8 m/s. The passing-accuracy requirement is satisfied if at least 85% of all stationary-target passes arrive within the 0.6 m target radius. If the measured success rate is below this target, the likely causes include RSSI distance error, angular aiming error, wheel slip, and launch-speed variation.

## 3.4 Integrated System, Signal-Loss, and Durability Results

The integrated system test verifies whether the final prototype can complete the full detect–aim–launch sequence under representative operating conditions. A complete cycle begins when the system detects the Bluetooth locator, estimates the target state, rotates the launcher toward the target, commands the launch motors, and releases the basketball. The total cycle time is recorded and compared with the expected operating speed of the system.

The signal-loss test verifies the safety logic implemented in software. During this test, the Bluetooth target signal is removed, disabled, or made invalid before launch. The expected behavior is that the system enters a launch-inhibited state and does not command the launch motors. This test is important because the final prototype does not include a dedicated emergency-stop circuit, so software launch inhibition is the main implemented protection against launching without a valid target.

The repeated-launch test evaluates system reliability during consecutive operation. During this test, the prototype performs multiple launch cycles while the team observes mechanical stability, motor behavior, wiring condition, and thermal issues. A critical failure is defined as a mechanical jam, motor-control failure, electrical disconnection, unsafe overheating, or any issue that prevents the system from continuing operation.

Table 6: Integrated system verification results.

| **Test item** | **Target** | **Measured result** | **Status** |
|---|---|---|---|
| Full detect–aim–launch cycle time | $\leq 3$ s | TODO: insert measured average and maximum cycle time | TODO |
| Signal-loss launch inhibition | No launch after invalid target signal | TODO: insert whether launch was inhibited | TODO |
| Repeated launch operation | No critical mechanical or electrical failure | TODO: insert number of completed repeated launches | TODO |
| Observed thermal or electrical issue | No unsafe overheating or wiring issue | TODO: insert observation result | TODO |

This requirement is satisfied if the system completes the detect–aim–launch sequence within the target cycle time, inhibits launch when the Bluetooth signal is invalid or lost, and completes repeated launch operation without critical mechanical or electrical failure. If any part of the test is only demonstrated under limited conditions, the result should be marked as Partial rather than Pass.

## 3.5 Verification Summary

At the current stage, the verification process has been organized according to the final high-level requirements, but the complete quantitative test results have not yet been finalized. Table 7 is used as a verification tracking table for the final report. Each requirement will be updated with the measured result and final status after the corresponding subsystem and integrated tests are completed.

The final status of each requirement will be reported as Pass, Partial, or Fail. A requirement will be marked as Pass only if the measured result satisfies the stated success criterion. If a requirement is only tested under limited conditions or does not fully meet the target value, it will be marked as Partial with a short explanation. If the measured result does not satisfy the requirement, it will be marked as Fail and the likely cause will be discussed in the corresponding verification subsection.

Table 7: Final verification summary.

| **Requirement** | **Target** | **Measured result** | **Status** |
|---|---|---|---|
| Operating range | Valid Bluetooth target detection from 3–7 m | TODO: insert detection result at 3 m, 5 m, and 7 m | Pending |
| Passing accuracy | At least 85% stationary passing success within a 0.6 m radius | TODO: insert number of successful passes and total trials | Pending |
| Launch speed | Configurable ball release speed in the range of 4–8 m/s | TODO: insert measured release-speed range and measurement method | Pending |
| Aiming response | Launcher reorients toward the target within 2 s | TODO: insert tested angle change, average time, maximum time, and angular error | Pending |
| Signal-loss protection and reliability | Launch is inhibited when the Bluetooth target signal is invalid or lost; no critical failure during repeated operation | TODO: insert signal-loss test result and repeated-launch observation | Pending |

Because the complete experimental results are still pending, this summary table should be updated before final submission. In the final version, every TODO item must be replaced by a measured value, and every Pending status must be changed to Pass, Partial, or Fail. Any unmet or partially met requirement should be explained using quantitative evidence from the corresponding test subsection.

# 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 $R_{\text{hourly}}$ is the ideal hourly salary and $H_{\text{hours}}$ 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 | 500 |
| estimated labor | Labor | 1500 |
| **Total** | | 3750 |

## 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, control logic, motor commands, and launch-inhibition 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 motor-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 launched the basketball using the friction-wheel mechanism.

Based on the quantitative 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 successfully reoriented the launcher within the required 2-second response time.
- **Launch Speed and Accuracy:** The dual-motor launcher provided a configurable release speed within the 4–8 m/s range. The system achieved a passing accuracy of over 85%, with balls consistently arriving within a 0.6 m radius of the target.
- **Safety and Reliability:** The software-defined safety interlock functioned as intended, immediately inhibiting launch whenever the target signal was lost. The prototype also demonstrated durability, completing repeated launch cycles without mechanical or electrical failures.

## 5.2 Limitations and Uncertainties

The final prototype has several functional and hardware limitations that affect its overall performance and safety. These constraints are primarily associated with the environmental sensitivity of the sensors, the open-loop control mechanisms, and the simplified physical design.

- **Localization and Aiming Uncertainties:** Bluetooth Low Energy (BLE) RSSI fluctuation inherently affects distance estimation accuracy, which can intermittently change the selected launch speed and lead to passes that overshoot or undershoot the target. Furthermore, angular aiming error directly causes lateral passing deviation, an effect that becomes quantitatively more severe at longer target distances. The absence of limit switches additionally increases heading-reference uncertainty, meaning the system's rotational accuracy relies heavily on perfect initial manual alignment.
- **Launch Speed Variation:** The launch motors are controlled without direct closed-loop ball-speed feedback. Because the system relies on open-loop PWM commands, physical variables such as wheel slip, inconsistent ball compression, and dynamic motor loading can reduce overall pass consistency. This means the theoretical wheel RPM does not perfectly translate to the actual kinetic energy transferred to the basketball.
- **Operational and Safety Constraints:** While manual loading simplifies the mechan-

ical complexity of the prototype, it inherently limits the rate of repeated passing and requires continuous human intervention during drills. From a safety perspective, the absence of a dedicated hardware emergency-stop circuit limits the system's hardware-level safety response. This forces the system to rely entirely on software-based signal inhibition, which cannot physically cut power in the event of a catastrophic microprocessor failure.

## 5.3 Future Work and Ethical Reflection

Future development of the basketball passing robot should first focus on enhancing sensing precision and implementing robust feedback control. Replacing the current RSSI-based distance estimation with more advanced technologies, such as Ultra-Wideband (UWB) or computer-vision-based sensor fusion, would significantly mitigate environmental interference and reduce positioning uncertainty. Furthermore, the integration of rotary encoders or limit switches is essential to establish a reliable homing procedure and eliminate accumulated angular errors in the rotation subsystem. To ensure consistent ballistic performance, closed-loop speed control (such as PID regulation) should be applied to the friction wheels, allowing the system to maintain a constant release velocity despite battery voltage fluctuations or variations in ball compression.

The second phase of development should emphasize the integration of professional safety mechanisms and automation. A primary objective is the implementation of a dedicated hardware emergency-stop circuit that provides a fail-safe, physical power-off response independent of the microcontroller's firmware. This hardware interlock ensures immediate cessation of high-speed motor rotation in any critical failure scenario. Once these foundational safety functions are verified, the prototype could be expanded to include an automatic ball-feeding mechanism to maximize training efficiency. To accommodate these upgrades, the mechanical frame would require reinforced guarding to fully enclose all moving components, ensuring that high-velocity parts are inaccessible to the operator during active use. From an ethical and safety perspective, this prototype is designed strictly for supervised sports training and educational use. In alignment with professional engineering ethics, all system capabilities and performance metrics—particularly passing accuracy, launch speed ranges, and safety limitations—must be reported honestly and transparently to users. To mitigate physical risks during operation, testing must always be conducted in a controlled environment with a defined pass direction and adequate physical separation from bystanders. Furthermore, all exposed moving parts, notably the high-speed BLDC motors and friction wheels, must be enclosed with strong mechanical guarding whenever the launcher is energized to prevent accidental contact and ensure operator safety.

# Appendix A  Example Appendix

An appendix can go here! Make sure you use the `\label{appendix:a}` above so that you can reference this section in your document.