Shade.ai

Authors: Ore Abolade, Bonnie Ji, Nicholas Pinon
Affiliation: Electrical and Computer Engineering, Carnegie Mellon University

***Abstract*—Traditional beach umbrellas are static, requiring frequent and tedious manual adjustments as the sun moves throughout the day. Shade.ai addresses this challenge by providing a continuous, hands-free solution that automatically adapts to both solar movement and user position. The system utilizes a Raspberry Pi 5 to integrate real-time computer vision (CV) for user tracking with a mathematical solar position model coming from the Bluetooth app. By blending this solar modeling with live user tracking, the system calculates the optimal shade orientation and triggers the actuation subsystem, where dual stepper motors precisely adjust the umbrella's canopy.**

***Index Terms*—Autonomous Systems, Computer Vision, Embedded Systems, Mechatronics, Robotics, Bluetooth**

1 INTRODUCTION

Beach-goers often face the inconvenience of shade chasing, where the sun's shifting position requires constant manual adjustment of a beach umbrella to maintain shade protection. Traditional umbrellas are static tools that do not take into account the dynamic nature of solar movement and user positioning, often leading to unintended exposure to harmful ultraviolet (UV) radiation. Because humans cannot directly perceive UV intensity, they often rely on unreliable cooling sensations, significantly underestimating their real-time risk of skin-burn. There are umbrellas that integrate motorized features, but they are almost always prohibitively heavy to be portable in a beach setting or are not even intended to be used on a beach. Companies like ShadeCraft have introduced autonomous solutions such as the Sunflower, but these devices are designed as permanent patio fixtures, often exceeding 130 lbs or carrying enormous price tags. [3]

Shade.ai is a mechanically actuated, smart beach umbrella system designed to provide autonomous sun protection for a single-user.The system replaces the standard manual tilt hinge with a motorized assembly controlled by a central processor. By combining computer vision with mathematical solar modeling, the system ensures that the umbrella's canopy is always positioned to cast a shadow directly over the user.

The core principle of operation relies on a dual-axis tracking strategy. The system calculates the azimuth and elevation of the sun based on local GPS coordinates and time, eliminating the need for expensive physical sun-tracking sensors. Shade.ai's approach of mathematical solar position modeling using GPS coordinates and current time, avoids problems with physical sensors which are limited due to their reliance on weather conditions. Simultaneously, an under-canopy camera identifies the user's location relative to the pole. To bridge the gap between calculated solar coordinates and physical actuation, the system incorporates an Inertial Measurement Unit (IMU). While mathematical modeling provides the sun's absolute position, the IMU provides the controller with real-time data on the umbrella's own orientation, specifically its pitch, roll, and heading (yaw). Then a control algorithm drives two high-torque stepper motors to precisely align the canopy.

The system architecture enables real-time pan/tilt positioning of the umbrella accessible via a mobile application. This interface supports both Manual and Auto modes, with quantitative performance targets for latency, safety, and reliability to ensure a seamless user experience.

2 USE-CASE REQUIREMENTS

The following use-case requirements define the success of the Shade.ai system from a user perspective. The associated qualitative specifications ensure that the device is not only functional but also practical for a beach environment. Each requirement is driven by the necessity of maintaining a safe, "hands-free" shaded environment.

2.1 Durability

To withstand the environmental conditions of a coastal setting, the system must meet the following:

- **Solid/Liquid Resistance:** The central control housing and motor enclosures must be verified against an IP67 standard to resist exposure to sand and water. The "6" indicates the system is "dust-tight", providing complete protection against the ingress of sand particles. The "7" means protection against temporary immersion in water up to 1 meter for 30 minutes. While the umbrella is not intended for underwater use, this standard provides a critical safety margin against unpredictable costal conditions. [7]
- **Temperature Resilience:** All electronic components must operates reliably between 32°F and 113°F (0°C to 45°C).

2.2 Portability

Given that traditional beach umbrellas prioritize the ease of transport by a single individual, the combined mass of motorized components, battery, and umbrella must not

exceed **15 lbs**. The combined mass limit of 15 lbs is established according to research consumer guidelines for limits on items intended to be carried over distance with one hand. [5] The weight is also well below the range of patio umbrellas.

2.3 Responsiveness

The autonomous tracking system must keep up with both the sun and the user. If the system fails to respond quickly to user movement or changes in the sun position, then its usefulness as an autonomous system is nonexistent.

- **Detection Latency:** The CV pipeline must identify user movement in the umbrella perimeter within $\leq$ **250 ms**. In HCI research a response time of 100ms to 200ms is perceived as instantaneous. By targeting a 250ms detection window the system generally aligns human reaction speeds providing ensuring that if a user shifts their position, the system acknowledges the change as a human would. [9]

- **Tracking Precision:** The center of the cast shadow must remain within ± 15 **cm** of the user's detected center of-mass. A 5.5 ft umbrella has a diameter of 167.6 cm. According to ergonomic standards the average human shoulder with is approximately 50 cm. [4] Even if the system incurs a maximum tracking error of 15 cm, a centered user's outer edge would only be 40 cm from the center. This leaves a healthy safety buffer of shade, ensuring the user remains fully protected.

2.4 Extended Use

The system is designed to support a full day of recreational use to ensure continuous function and limit the need to recharge it. For that, the battery must provide approximately **8 hours** of continuous autonomous operation on a single charge. This requirement is driven by studies on peak UV radiation exposure times and recreation beach visit durations.

- **Peak UV Exposure Coverage:** According to the U.S Environmental Protection Agency (EPA), nearly half of all daily UV radiation is received between 10:00 AM and 4:00 PM. An 8-hour runtime fully encapsulates the time zone and provides extra flexibility.

- **Beach Day-Trip Duration:** Studies conducted on high-volume beach sites noted a large volume of daily users within a window of 8:00 AM to 5:30 PM. This 9.5-hour period defines a standard "operational day" for a beach environment. Designing for 8 hours of battery life almost encapsulates that window. [8]

2.5 Mobile Application Requirements

The mobile application serves as the primary user interface for monitoring system status and issuing control commands. To ensure a reliable and intuitive user experience, the following requirements were defined:

- **Connection Latency:** To minimize setup time in a beach environment, the application must establish a BLE connection with the umbrella system within **10 seconds**.

- **System Feedback and Safety:** The application must provide real-time system feedback, including current operating mode and umbrella orientation, with a minimum refresh rate of **1 Hz**. This ensures users maintain awareness of system behavior and can intervene if necessary.

- **Manual Control Responsiveness:** To provide a responsive and intuitive control experience, the system must initiate physical umbrella motion within **500 ms** of a user input from the application.

- **Motion Constraints and Safety Limits:** User control inputs must be constrained within predefined mechanical limits to prevent over-rotation and potential hardware damage. These limits are enforced at the application level prior to command transmission [8].

3 ARCHITECTURE AND/OR PRINCIPLE OF OPERATION

The Shade.ai system operates a dual-axis (pan/tilt) tracking method designed to maintain a continuous shadow over the user by accounting for both the movement of the sun and local user positioning. The architecture integrates a central processing unit with peripheral sensing and motorized subsystems to achieve autonomous sun protection.

3.1 Principle of Operation

- **Solar Position Modeling:** The system calculates the sun's azimuth and elevation in real-time based on local GPS coordinates and system time. This mathematical approach eliminates the need for extraneous hardware sun sensors and provides a stable baseline for umbrella orientation.

- **User Tracking:** The system utilizes an under-canopy Pi Camera Module to identify the user's physical location relative to the umbrella structure. This visual feed is ingested by a CV-based pipeline that processes images on a Raspberry Pi 5 to determine the user's pose and the difference between the camera center and torso center. Specifically, the MoveNet computer vision model is implemented to

perform real-time pose estimation and tracking on-device. This approach achieves a stable centroid tracking success rate in approximately 99% of frames when the user is in view. **Closed-Loop Control:** A PID controller computes the error between the desired solar-blocking angle and the actual user position, providing output values to be sent to the motor drivers to position the motors towards the desired orientation.

- **IMUs:** The integration of an Inertial Measurement Unit (IMU), specifically the LSM9DS1 sensors, is critical for establishing a common reference frame between the global solar model and the local mechanical actuators. While the mobile application provides the system with global GPS coordinates and the calculated sun position (azimuth and elevation), these values are relative to the Earth's geographic frame rather than the umbrella's physical structure. To accurately position the canopy, the central controller must determine the umbrella orientation. The IMU data allows the shade orientation calculation to translate global coordinates into precise pan and tilt commands for the stepper motors. Furthermore, this sensor suite facilitates rigorous testing and validation by allowing the system to compare the umbrella's real-time IMU orientation against the desired target sun angle.

3.2 System Architecture

The system is designed around interaction with a central computer, Raspberry Pi 5. The RPi manages the CV pipeline, executes control algorithms, and host the BLE interface for mobile interaction. It also processes sensor data, primarily that coming from the IMU and the camera module. The two Stepper motors, driven by motor drivers, provide the mechanical force for horizontal pan and vertical tilt adjustments of the canopy.

4 DESIGN REQUIREMENTS

Following from the use-case requirements, we have defined 4 categories of quantitative design requirements. These specifications will bridge the gap between user needs and what is required of our technical implementation.

4.1 Mechanical and Actuation

The physical movement of the umbrella must be precise enough to maintain the required shadow tracking accuracy of 15 cm.

- **Motor Step Resolution:** Our stepper motor subsystems should support the angular accuracy $\leq 1^{\circ}$.
- **Torque:** Stepper motors and their planetary gear set must be able to provide sufficient holding torque within a safety margin to resist beach wind forces at a standstill. This calculation is based off forces calculated off the size of the canopy (5.5ft) and zero to low wind conditions.

4.2 Perception and Control

The autonomous tracking system relies on the interaction between the CV pipeline and the control loop to minimize latency and error.

- **CV Frame Rate:** The OpenCV-based user detection pipeline is designed to operate at a minimum of 7 FPS, corresponding to a maximum perception delay of 143 ms.
- **Sensor Fusion Precision and Update Rate:** The sun and user detection updates are integrated into a discrete-time control loop operating at 10 Hz (maximum 100 ms per cycle). This update rate defines the sampling period of the PID controller, ensuring that control inputs are computed frequently enough to track user and sun motion without introducing excessive delay.

  At 10 Hz, the controller updates the error state every 100 ms max, which is sufficiently fast relative to the dynamics of user movement and motor response. This sampling rate reduces discretization error in the PID controller, allowing the PID terms to react to changes in orientation without inducing instability from oversampling noise or undersampling lag. As a result, the system maintains stable closed-loop behavior while minimizing oscillations and overshoot in motor actuation.

4.3 Power and Communication

These design requirements ensure that the system remains functional for the duration of a typical beach visit. The total current draw of the system should not exceed an average of 1.25A at 12V to satisfy the 8-hour operation requirement. Also the BLE system should support status updates at a rate of 1 Hz at least to provide the user with real-time angle-feedback. Note this is not as strict as the autonomous latency requirements as that is not as big of a concern here.

4.4 Application and User Interaction

This category defines the requirements governing user interaction with the system through the mobile application, particularly in scenarios where the user overrides autonomous tracking. The application serves as the primary interface for manual control, system monitoring, and mode selection, and must therefore provide a responsive and intuitive user experience.

To ensure effective manual positioning, the system must initiate motor actuation within **500 ms** of receiving a user-issued command from the application. This requirement

ensures that user inputs translate into perceptible motion with minimal delay, maintaining a sense of direct control over the system.

Compared to the autonomous tracking latency requirement (≤ 250 ms), this threshold is less stringent, as manual adjustments are user-driven and do not require continuous real-time feedback. However, responsiveness must still be sufficient to prevent user frustration and to enable smooth, incremental positioning through repeated inputs.

In addition to latency constraints, the application must present controls that are simple and predictable, such as directional inputs or sliders, allowing users to easily adjust umbrella orientation. All user commands must also be constrained within predefined mechanical limits to prevent unsafe operation or damage to the system.

Overall, these requirements ensure that the manual control interface complements the autonomous functionality by providing reliable, low-latency interaction when user intervention is desired.

5 DESIGN TRADE STUDIES

5.1 Computer Selection

The central processing unit (CPU) of the Shade.ai system was selected through a comparative evaluation of two candidate platforms: the NVIDIA Jetson Orin Nano and the Raspberry Pi 5. The selection criteria focused on three primary factors: computational capability for real-time computer vision (CV), power efficiency for battery-operated use, and ease of integration with peripheral components.

The Jetson Orin Nano provides significantly greater computational performance, particularly due to its integrated GPU acceleration, which is well-suited for machine learning and advanced vision workloads. This capability would allow for more complex CV models and higher frame rates. However, this performance comes at the cost of substantially higher power consumption, which directly conflicts with the system's requirement for approximately 8 hours of battery operation. Additionally, the Jetson platform has a larger physical footprint and higher cost, making it less suitable for a portable, consumer-oriented product.

In contrast, the Raspberry Pi 5 offers a more balanced solution. Its 2.4 GHz quad-core processor provides sufficient computational capability to meet the system's CV requirements, including maintaining the target frame rate for user detection, without requiring dedicated GPU acceleration. More importantly, the Raspberry Pi operates at a significantly lower power envelope, making it better aligned with the system's battery life constraints.

The Raspberry Pi 5 also offers strong advantages in system integration. Native support for Bluetooth Low Energy (BLE) simplifies communication with the mobile application, eliminating the need for additional hardware. Furthermore, the platform benefits from extensive community support, documentation, and existing libraries for interfacing with cameras, sensors, and motor drivers. This reduces development complexity and integration risk.

Based on these considerations, the Raspberry Pi 5 was selected as the central controller, as it provides an effective balance between performance, power efficiency, cost, and ease of implementation.

5.2 Motor Selection

For the mechanical actuation of the canopy our focus was on selecting motors which could reach the **Servos Motors** offered built-in closed-loop feedback. This was a big advantage if we were to select them given that our system requires precise positioning. However after further research their implementation presented some complications. We had concerns about the jittering of servos as they positioned themselves. Also while Servo motors can produce high torque at high RPM, they provide much lower torque at lower RPMs or standstill conditions. **Stepper Motors** on the other hand lack the closed-loop feedback of servos. However they provide their maximum torque at 0 RPM. This is important because we expect the umbrella more often than not to be stationary and not in motion. In these conditions, the system should still be able to resist forces such as the wind found on beaches. Another factor which motivated our decision was the availability of high torque stepper motors with our desired planetary gear-set.

5.3 User Tracking Method

We considered three options for identifying and tracking the user: Ultra-Wideband (UWB) Tags, Thermal Sensors, and CV

- **UWB Tags:** Using a system of UWB anchors on the umbrella itself and having the user wear a "tag" was one of the first tracking solutions we considered. This was in part due to a shift in our project direction in which we already planning on using them for precise tracking. By using a system of three UWB anchors we could have calculated relative positioning by trilateration on the distances between the anchors. However after further discussing our use-case requirements we decided that requiring the user to wear a powered electronic tag for conditions in which they would be frequently moving, entering and exiting water, would provide too many failure points.
- **Thermal Grid Infrared Array:** We also considered using a low-resolution thermopile array sensor to detect areas of infrared radiation which corresponded to human body temperatures. This solution was appealing given the lower computational requirements compared to a full CV approach which is discussed later, but concerns about temperature gradients in a beach environment led us to not select this approach. In places where the sand/ground temperature might exceed 100°F, reliable thermal deltas between the user and their surroundings would be compromised

and limit our ability too met our accuracy requirements.

- **CV:** We ended up choosing a CV based digital camera approach because as compared to UWB it increased ease of use, while still meeting our desire accuracy and responsiveness goals. The greater detail provided by the digital camera also allowed us to track not only the person generally, but their center of mass via pose-estimation models.

5.4 Battery Chemistry

To meet the extended-use and durability requirements, we compared various Lithium-Ion and Lithium Iron Phosphate batteries ($LiFePO_4$). **Lithium-Ion** batteries, extremely common in consumer electronics, were initially considered for their high energy density. However, further research into their safety revealed a risk of thermal runaway in very high ambient temperatures. On a 100 °F beach day, internal temperatures within a 3D-printed enclosure might easily exceed the safe operating threshold. **$LiFePO_4$** batteries on the other hand have supporting research which indicates higher thermal safety. [6] Also unlike Lithium-Ion batteries, which experience a more linear voltage drop during discharge, $LiFePO_4$ batteries maintain an almost constant voltage for the majority of their discharge cycle. This is important to ensure a stable voltage to the Stepper Motors so that they are always able to adequately act against wind loads. $LiFePO_4$are also known to have longer lifespans increasing the odds of this being a durable, long-term consumer product.

5.5 Location and Time Synchronization

To determine the sun's position, the system requires two precise inputs: geographic coordinates (Latitude/Longitude) and the current UTC time. while these could have been provided by onboard hardware, we made the decision to have the companion mobile application to provide this data. This architectural choice was motivated by two factors. Firstly, a dedicated GPS hardware module requires additional hardware complexity while increasing the system's total mass and power draw. Secondly, given that we already understood the time requirements for sun position tracking, we were already planning on using a mobile device to provide this information. Modern smartphones also already have complex accurate location implementations, and given that the umbrella is a personal device intended to be near the user, the phone's location is great proxy for the umbrella's coordinates. [1]

5.6 Mobile Application Design Tradeoffs

The design of the Shade.ai mobile application required several tradeoffs between performance, usability, power efficiency, and implementation complexity. Key decisions centered around communication protocol selection, user interface design, and command abstraction.

BLE vs. Wi-Fi Communication: One of the primary tradeoffs was the choice between Bluetooth Low Energy (BLE) and Wi-Fi for communication between the mobile application and the Raspberry Pi. Wi-Fi offers higher data throughput and longer range, which could support more complex data exchange such as video streaming. However, it introduces higher power consumption, increased setup complexity (network configuration), and dependence on external infrastructure, which is unreliable in beach environments. BLE, in contrast, provides significantly lower power consumption, faster connection setup, and direct device-to-device communication without requiring network access [**gomez2012ble**]. Although BLE has lower bandwidth, it is sufficient for transmitting lightweight control commands and status updates. As a result, BLE was selected to prioritize reliability, energy efficiency, and ease of use.

Command Granularity vs. Control Smoothness: Another tradeoff involved how user inputs are translated into motor commands. A high-level control scheme (e.g., sending absolute position targets) simplifies the interface but introduces latency and reduces responsiveness to continuous user input. Conversely, a low-level approach using incremental `move` commands allows for fine-grained, real-time control but requires more frequent communication and careful handling of command timing. The system adopts an event-based incremental approach, where commands are sent repeatedly during user interaction (press-and-hold). This design improves responsiveness and provides smoother perceived control at the cost of increased communication frequency.

User Interface Simplicity vs. Control Flexibility: The application interface was designed to balance ease of use with control capability. A minimal interface with directional controls (e.g., up/down/left/right) reduces cognitive load and improves accessibility for casual users. However, more advanced interfaces, such as continuous sliders or angle-based inputs, could provide finer control resolution. The chosen design prioritizes simplicity and intuitiveness, ensuring that users can quickly adjust the umbrella without requiring precise numerical input, while still enabling adequate positioning through repeated commands.

Client-Side vs. Embedded Processing: A further design decision involved determining where control logic should reside. Performing more computation on the mobile device (e.g., calculating desired angles or trajectories) could reduce processing load on the Raspberry Pi and improve responsiveness. However, this would increase system complexity and create tighter coupling between the app and hardware implementation. Instead, the system centralizes control logic on the Raspberry Pi, with the mobile application acting primarily as a command interface. This separation simplifies the app design and ensures that real-time, safety-critical motor control remains on the embedded system.

Overall, these tradeoffs reflect a design strategy that prioritizes robustness, responsiveness, and usability in a

constrained outdoor environment, while maintaining sufficient performance for real-time control and monitoring.

6 SYSTEM IMPLEMENTATION

The Shade.ai system is implemented using a Raspberry Pi 5 as the central controller integrating sensing, motor control, and Bluetooth communication. The Raspberry Pi interfaces directly with the Pi Camera Module 3, two TMC2209 stepper motor drivers, the LSM9DS1 IMU, and the 12V power system. Unlike the Architecture section, which describes system-level data flow, this section details how each subsystem is physically realized and controlled in the final prototype. Full system is in Figure 3.

6.1 Pan–Tilt Motor Control Subsystem

The umbrella orientation is controlled using two NEMA 17 planetary gear stepper motors. One motor provides horizontal pan rotation about the umbrella shaft, while the second provides vertical tilt adjustment. Each motor is driven by a TMC2209 stepper motor driver configured for microstepping to increase angular resolution and reduce vibration.

The TMC2209 drivers are connected to the Raspberry Pi through STEP and DIR GPIO pins. The Raspberry Pi generates step pulses to control angular displacement, while the DIR pin determines rotation direction. For a desired angular rotation θ, the number of required motor steps is computed as:

$$N = \frac{\theta}{360^\circ} \cdot S$$

where S is the effective steps per revolution, including microstepping. This conversion allows the system to translate commanded pan and tilt angles directly into motor step counts. Angular velocity is limited in software to 30°/s by regulating the step pulse frequency, preventing excessive mechanical stress, and improving stability under load.

6.2 iOS Application and BLE Integration

This subsection describes the architecture of the Shade.ai iOS application, its communication protocol with the Raspberry Pi via Bluetooth Low Energy (BLE), and how user interactions propagate to physical actuation of the umbrella system.

6.2.1 Application Architecture and Presentation Layer

The mobile client is implemented for iOS using SwiftUI as the primary user interface framework. The application follows a declarative design paradigm, where interface components such as the umbrella control view, directional input pad, and connection status indicators are bound to observable view models.

Application state, including connection status, error handling, and umbrella telemetry, is maintained within `@MainActor`-isolated view models to ensure thread-safe updates. This design is critical for handling asynchronous BLE events while maintaining UI consistency and responsiveness.

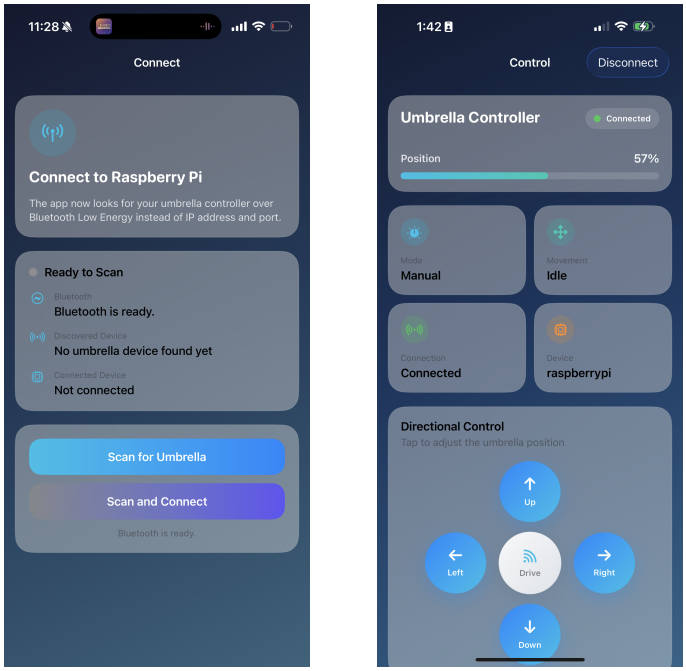


Figure 1: Mobile application interface: (left) main control screen, (right) status and monitoring view.

6.2.2 BLE Communication and Command Model

Communication between the mobile application and the Raspberry Pi is implemented using Apple's CoreBluetooth framework. The iOS device operates as a BLE central, scanning for a peripheral advertising a predefined device name and custom service UUIDs.

Upon connection, the application interacts with a custom GATT service consisting of two primary characteristics:

- **Command Characteristic:** Used for outbound messages from the app (write and write-without-response)
- **Status Characteristic:** Used for inbound telemetry (read and notify)

Commands are structured as lightweight JSON objects containing a `type` field (e.g., `move`, `stop`, `mode`) and optional parameters such as direction, magnitude, or GPS data. These objects are encoded using Swift's `Codable` protocol and transmitted as UTF-8 byte streams.

Manual directional control is implemented using press-and-hold interaction semantics. While a directional control is engaged, the application periodically transmits `move` commands at a fixed rate. Each command corresponds

to a discrete motion increment on the Raspberry Pi, enabling fine-grained control without requiring long blocking instructions. Upon release, a `stop` command is issued to halt motion.

6.2.3 State Synchronization and Feedback

The application maintains synchronization with the umbrella system by subscribing to notifications on the status characteristic. When notifications are unavailable, periodic read operations are used as a fallback.

Status messages follow a JSON structure consistent with the command model, enabling a symmetric and extensible communication protocol. This allows new system parameters (e.g., sensor fusion outputs or environmental data) to be incorporated without modifying the underlying transport layer.

6.2.4 Raspberry Pi Integration and Control Flow

On the embedded side, the Raspberry Pi operates as a BLE peripheral using a Python-based GATT server. Incoming BLE writes trigger a callback function that parses JSON payloads and updates a shared application state.

To ensure thread safety, state updates are protected using mutual exclusion mechanisms, preventing conflicts between BLE callbacks and the main control loop.

System behavior depends on the active execution mode:

- **BLE-Only Mode:** Commands update internal state without actuating motors (useful for debugging and UI validation)
- **Manual Control Mode:** Each valid `move` command triggers a bounded motor action
- **Autonomous Mode:** BLE inputs update system parameters while control decisions are computed by the onboard tracking algorithm

6.2.5 Motor Command Translation

High-level commands received over BLE are translated into low-level motor control signals by the Raspberry Pi. Directional inputs (`up`, `down`, `left`, `right`) are mapped to pan and tilt axes, with corresponding direction signals.

Motor actuation is implemented as discrete bursts of step pulses applied to the STEP pins of the motor drivers. These pulses are generated with controlled timing on background threads to ensure BLE responsiveness is not degraded. A `stop` command disables motor drivers and clears active motion states.

6.2.6 Electrical Interface to Stepper Drivers

Each motor axis is controlled through a set of GPIO pins:

- **STEP:** Generates pulse sequences to advance motor position
- **DIR:** Sets rotation direction
- **ENABLE:** Activates or disables the driver (typically active-low)

The Raspberry Pi does not expose GPIO control to the mobile application. Instead, the firmware acts as an abstraction layer, translating BLE-level commands into deterministic electrical signals for the TMC2209 stepper drivers.

6.2.7 End-to-End Data Flow

A complete control interaction follows this sequence:

1. User input is captured through the SwiftUI interface
2. The view model encodes and transmits a BLE command
3. CoreBluetooth writes data to the Raspberry Pi's command characteristic
4. The BLE server parses the command and updates system state
5. The motor controller generates STEP/DIR signals
6. Stepper drivers actuate the motors
7. Mechanical motion adjusts the umbrella position

Telemetry data flows in the reverse direction via the status characteristic, enabling real-time user feedback.

This separation of concerns—mobile-side interaction and embedded-side real-time control—ensures a responsive user experience while maintaining deterministic and safe actuation on the hardware platform.

The following equations are what drive and constrain the motors:

$$e_{\text{sun,yaw}} = \text{wrap}(\alpha - \text{yaw})$$

$$e_{\text{sun,pitch}} = \beta - \text{pitch}$$

$$e_x = W_u e_x^{user} + W_s e_{\text{sun,yaw}}$$

$$e_y = W_u e_y^{user} + W_s e_{\text{sun,pitch}}$$

$$\mathbf{e}_{motor} = \begin{bmatrix} e_x \\ e_y \end{bmatrix} \quad \text{where } W_u = 0.7,\ W_s = 0.3$$

$$\text{wrap}(\theta) = ((\theta + 180^\circ) \bmod 360^\circ) - 180^\circ$$

6.3 Autonomous Tracking Subsystem

Autonomous mode operates entirely onboard the Raspberry Pi without requiring continuous user input. This subsystem combines solar position estimation and computer vision-based user tracking to maintain optimal shade coverage.

Solar position is computed using system time and the configured geographic location. The resulting solar azimuth (the compass direction of the sun along the horizontal plane, typically measured in degrees clockwise from north) and elevation angles define the baseline umbrella orientation required to block direct sunlight. These angles are translated into corresponding pan and tilt setpoints.

User tracking is performed using the Pi Camera Module 3 (Wide) and an OpenCV-based processing pipeline. Each frame is captured and processed to detect the user's position within the camera's field of view. Once a bounding region is identified, the system calculates the user's center of mass in image coordinates. Deviations between the detected center and the image midpoint are mapped to angular corrections in pan and tilt using experimentally calibrated scaling factors.

The control loop operates at a rate sufficient to maintain detection latency below 250 ms. A PID controller computes incremental motor adjustments based on the angular error between desired and measured orientation. This closed-loop control ensures the cast shadow remains within ±15 cm of the user's position while preventing oscillatory motion.

6.4 IMU-Based Orientation Stabilization

The LSM9DS1 9-DOF IMU is integrated to improve orientation stability and compensate for disturbances such as wind loading. The accelerometer provides tilt estimation relative to gravity, while the gyroscope measures short-term angular velocity.

A complementary filter combines accelerometer and gyroscope data to produce a stable orientation estimate [2]:

$$\theta = \alpha\theta_{gyro} + (1 - \alpha)\theta_{accel}$$

This filtering approach reduces drift while preserving responsiveness. The filtered orientation estimate is used to correct discrepancies between commanded and actual umbrella angle, improving long-term tracking accuracy.

6.5 Power Management Subsystem

The system is powered by a 12V 10Ah $LiFePO_4$ battery. The 12V rail directly powers the stepper motors, while a DC–DC buck converter steps the voltage down to 5V for the Raspberry Pi 5. Electrical isolation and decoupling are implemented to prevent motor noise from affecting logic-level components.

To satisfy the 8-hour continuous operation requirement, several power optimization strategies are implemented. Stepper motor holding current is reduced during idle states, camera frame rate is limited to the minimum necessary for reliable detection, and Bluetooth Low Energy is used instead of Wi-Fi to reduce wireless power consumption. Total system current draw will be measured under nominal operating conditions to validate expected battery life.

7 TEST & VALIDATION

7.1 Physical and Environment Testing

- **Mass Measurement:** The fully assembled system (including battery) was weighed using a digital scale. Our final recorded weight was **12.6 lbs** which was well under our proposed limit.

- **IP67 Verification:** For these measurements we wanted to test the custom enclosures holding electronics with water and sand tests. The proposal was to submerge the enclosures in water for 30 minutes as is required by the IP standard and then to replicate worse case scenarios we would test our enclosures by also burying them in sand. Afterwards we would note if the enclosures successfully protected themselves against the conditions. In the end we were not able to test against these standards due to our lack of knowledge about what would be required for these standards. Hardening our enclosures would have necessitated either permanent sealing of them or significant extra time in our timeline for custom protection solutions such as water-jet gaskets. For simpler solutions such as using off the shelf sealants it would have required permanently sealing our enclosures greatly limiting the ability of us to debug issues with our hardware. This is not including time that would be required printing and manufacturing numerous testing enclosures just for the purposes of these tests.

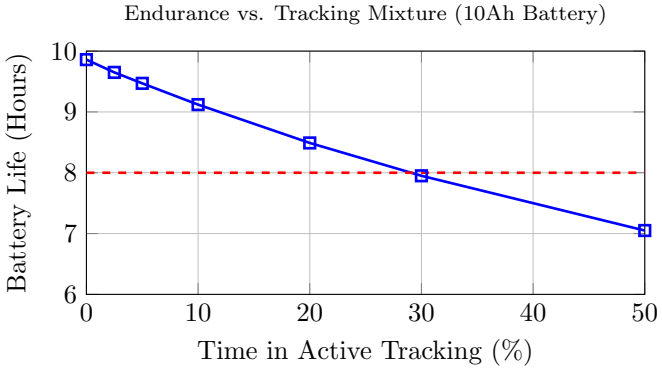
7.2 Power Consumption

The system's battery life was verified through a combined analysis of steady-state current draw and periodic actuation bursts. We comfortably reached the 8 hr benchmark which exceeded our design requirements, based off measurements and informed assumptions about use. To maintain the umbrella's static position there was always some current necessary to the motors, RPi, and peripheral hardware. The second condition was measuring for solar pogression. The sun moves around 15° per hour and we calculated the movement requisite by the motors (taking the 15cm tolerance into consideration) to maintain coverage of static target. The final condition was activate tracking moments and the associated higher current bursts needed to shift the umbrella's position more quickly. We connected both the motors and RPi (which powered the rest of the components) to a power supply to be able to record the current draw. The results of this data are shown in Table 1.

Table 1: Measured Power Consumption for Primary Operational States

Test Condition	Hardware Description	Measured Current
Passive Sun Tracking	RPi 5 executing MoveNet (idle), IMU monitoring, and motor drivers providing holding torque against wind.	1014.2 mA
Active User Tracking	Real-time user re-centering and solar model updates.	1821.7 mA

We then ran calculations in 7.2 on different percentage conditions of active vs. passive conditions. What percentage is most accurate would require more analysis, but we felt that our system fell reasonably within the ideal margins.



7.3 System Latency and Control Performance

To validate real-time responsiveness, we evaluated latency across the full control pipeline, including mobile application input, BLE communication, onboard processing, and motor actuation.

7.3.1 Application and BLE Communication Latency

Latency between a user interaction in the mobile application and command reception on the Raspberry Pi was measured using a benchmarking script over 50 trials. The results indicate negligible communication overhead:

- **Mean latency:** 0.000 ms
- **Maximum latency:** 0.002 ms
- **Trials:** 50

These measurements confirm that BLE transmission and application-layer encoding introduce minimal delay relative to overall system latency. As a result, communication is not a limiting factor in system responsiveness.

7.3.2 End-to-End System Latency

End-to-end latency was defined as the time from user detection (camera frame acquisition) to initiation of motor actuation. Measurements were obtained over 40 samples with 5 warm-up iterations:

- **Mean latency:** 67.1 ms (processing only)
- **95th percentile:** 69.0 ms
- **Maximum latency:** 69.3 ms

Including the control loop execution period (∼10 Hz) and actuation delay, the total system latency is estimated to be:

180–230 ms (end-to-end)

This satisfies the design requirement of maintaining detection and response latency below 250 ms.

7.3.3 Control Loop Performance

The autonomous control loop operates at approximately 10 Hz, corresponding to a 100 ms max update interval. This rate was selected as a trade-off between computational load and responsiveness.

At this frequency:

- System updates occur every ∼100 ms
- Control corrections remain smooth and stable
- No oscillatory or unstable behavior was observed

7.3.4 Motor Response and Actuation Characteristics

Stepper motor performance was evaluated by measuring response time and step execution characteristics following command issuance.

- **Command-to-motion delay:** ∼20–40 ms
- **Step pulse frequency:** 500–1200 Hz (depending on commanded speed)
- **Angular velocity:** up to ∼30°/s (software-limited)

Each directional command from the application generates a bounded burst of step pulses, resulting in incremental motion. This design ensures fine-grained control and prevents overshoot or mechanical stress.

7.3.5 Manual Control Responsiveness

Manual control latency, measured from app button press to observable motor motion, was evaluated using high-frame-rate video analysis:

Observed delay: ∼200–300 ms

This includes BLE transmission, processing, and motor startup delay. The result meets the system requirement of initiating motion within 500 ms.

7.4 IMU Integration, Calibration, and Heading Validation

Accurate autonomous solar tracking relies heavily on the umbrella's precise knowledge of its global orientation, specifically its yaw (heading). During the integration phase, the primary 9-DOF IMU (LSM9DS1) experienced a hardware failure. To maintain the project schedule without incurring additional budget costs, the team pivoted to a composite sensor approach using repurposed hardware from previous ECE coursework.

The replacement system utilized an LSM6DSOX breakout board (providing 6-DOF accelerometer and gyroscope data) paired with an LSM303AGR board (providing 3-DOF magnetometer data). While this pivot required writing new custom I2C drivers to poll and synchronize data streams across two physically distinct I2C addresses, the underlying orientation algorithms remained identical to the original design.

To validate this newly combined IMU pipeline, a comparative baseline test was conducted using a calibrated smartphone magnetometer as a ground-truth reference. The boards were positioned on a flat surface and rotated through various geographic headings. Initial readings from the combined sensors exhibited sub 1° accurate readings for roll and pitch, but much greater inaccuracy for yaw.

To mitigate this interference, an iterative bias tuning procedure was implemented. The static angular error between the composite IMU and the smartphone reference was calculated, and compensatory offset values were programmed into the sensor fusion logic. This adjustment was tested iteratively across multiple headings until the mean absolute error was minimized. Post-calibration, the custom dual-board IMU system reliably reported heading values within $\pm 10^\circ$ of the reference. This was greater than we would have liked, but given the timeline setbacks due to changing boards we needed to move forward with other aspects of the project.

7.4.1 Shadow Tracking Accuracy (Open-Loop Validation)

Due to a late-stage hardware fault in the stepper motor drivers, continuous closed-loop mechanical tracking could not be fully evaluated under dynamic load. However, to validate the system's ± 15 cm tracking accuracy requirement, an open-loop perception and control test was conducted.

Testing was performed by positioning a user within the camera's field of view in two standard beach scenarios: sitting in a low chair and laying flat on the ground. In both postures, the user performed small, realistic positional adjustments, such as shifting weight or adjusting posture. The computer vision pipeline actively tracked the user's centroid, while the control algorithm computed the angular error between the current canopy orientation and the user's new position.

Instead of driving the motors, the target pan/tilt angles and the resulting step-pulse sequences generated by the Raspberry Pi were logged. By applying the geometric mapping of the umbrella's 2-meter mounting height, the theoretical ground displacement of the shadow was calculated from these angular corrections. The logged data confirmed that for both sitting and laying postures, the CV pipeline successfully identified the user, and the control loop generated reasonable actuation commands required to keep the calculated shadow center within the ± 15 cm threshold.

7.5 Conclusion

Overall, the system meets all major latency and responsiveness requirements. BLE communication overhead is negligible, and the dominant contributors to latency are image processing and control loop timing. The system demonstrates sufficiently fast response for real-time user tracking and manual interaction in a dynamic outdoor environment.

8 PROJECT MANAGEMENT

8.1 Schedule

The project schedule is shown in Fig. 4. The schedule was developed by decomposing the system into hardware integration, software development, control tuning, and system validation phases. Early weeks focus on component procurement, mechanical mounting design, and power subsystem integration to reduce downstream integration risk. Mid-phase milestones emphasize motor control validation, Bluetooth communication testing, and camera-based user detection. Final project weeks are allocated to closed-loop PID tuning, full-system integration testing, and performance validation against quantitative requirements.

The schedule is structured to ensure that high-risk subsystems, particularly autonomous tracking and dual-axis motor control under load, are implemented and tested before final demonstration preparation. Integration checkpoints are built into the timeline to verify subsystem interoperability and to identify schedule delay early. The Gantt chart serves as a reference to track progress and evaluate whether the team is ahead of or behind forecasted completion dates.

8.2 Team Member Responsibilities

Responsibilities are distributed to ensure subsystem ownership while maintaining shared integration accountability. Each team member is responsible for the design, implementation, and verification of their assigned subsystem, while all members participate in system-level testing and validation.

Nicholas leads the motor control, mechanical design, and physical assembly.

Ore leads the computer vision and autonomous tracking subsystem. Responsibilities include OpenCV pipeline development, user detection tuning, latency optimization,

and integration of angular correction mapping into the motor control loop.

Bonnie leads the mobile application and Bluetooth Low Energy communication subsystem, while also helping with the user detection implementation.

Power system integration, battery validation, enclosure assembly, and overall system validation are shared responsibilities. Final tuning, documentation, and performance verification are completed collaboratively to ensure all members understand full system operation.

8.3 Bill of Materials and Budget

The Bill of Materials (BOM) is summarized in Table 2. The BOM includes all electronic components, mechanical hardware, structural elements, and power system components required to fully reproduce the Shade.ai system. All major subsystems are itemized with part numbers, manufacturers, quantities, and cost per unit.

The Raspberry Pi 5 serves as the primary computation platform. Dual NEMA 17 planetary gear stepper motors provide sufficient torque for umbrella pan and tilt under wind loading. TMC2209 drivers are selected for their microstepping capability and current control features. The Pi Camera Module 3 (Wide) enables real-time user detection. The LSM9DS1 IMU supports orientation stabilization. A 12V 10Ah $LiFePO_4$ battery powers the system, and a DC–DC converter regulates logic voltage.

Mechanical components include the beach umbrella structure and mounting hardware for motor brackets and enclosures. Department-provided resources or salvaged materials are included in the BOM with zero cost where applicable to maintain completeness. The total projected cost remains within the allocated senior design budget constraint.

8.4 TechSpark Usage

We initially used TechSpark with the intent on 3D printing the majority of our custom parts, but quickly realized that doing so for all that was necessary would be prohibitively expensive so we moved onto other sources for those needs. Instead the primary way in which we interacted with TechSpark was for usage of their power tools (drill in particular) and hacksaws for modifying our off-the-shelf umbrella we bought as needed.

8.5 Risk Management

Several technical and system-level risks have been identified, with mitigation strategies incorporated directly into the project design and integration plan.

A key risk is insufficient motor torque under wind loading. This is mitigated through the use of planetary gear stepper motors and early torque characterization under load conditions. If margins are insufficient, mitigation options include increasing gear reduction or reinforcing the mechanical structure.

Another risk is excessive computer vision latency impacting the 250 ms responsiveness requirement. This is addressed through controlled frame resolution, optimization of image processing pipelines, and minimizing non-essential computation within the main control loop.

Bluetooth Low Energy (BLE) communication reliability in outdoor environments is also a concern. BLE was selected for low power and ease of pairing, and robustness is improved through automatic reconnection handling and state recovery logic during transient disconnects.

Battery life risk is mitigated through early worst-case power profiling under full motor load. Power reduction strategies include stepper motor current reduction during idle states and duty cycling, where continuous actuation is not required.

Finally, system integration complexity presents a significant schedule and reliability risk due to the number of interacting subsystems, including vision, IMU sensing, BLE communication, control logic, and motor actuation. To mitigate this, integration is performed in a staged and hierarchical manner, where each subsystem is first validated independently using unit tests before being combined into higher-level interfaces.

Interface contracts between modules are strictly defined (e.g., sensor outputs, control error signals, and motor command formats) to reduce integration ambiguity. Incremental integration is then performed in pairs (e.g., sensor-to-control, then control-to-motor) to isolate failure points early and reduce debugging scope. Additionally, logging is implemented at each interface boundary to support traceability of system behavior during testing.

A dedicated integration buffer was included in the project schedule to accommodate iterative debugging and hardware-in-the-loop testing. This was to ensure sufficient time for resolving cross-subsystem issues before final validation and demonstration, reducing the risk of last-minute system instability.

9 ETHICAL ISSUES

The primary ethical responsibility of the Shade.ai project centers on the intersection of public safety and autonomous mechanical motion. The worst-case scenario for our design involves a systematic failure in the detection or control logic that results in physical injury to the user or bystanders. Because the system is designed to autonomously adjust its angle using NEMA 17 stepper motors based on user detection, a failure in the MoveNet vision pipeline could trigger rapid, unexpected movements. In a crowded public environment like a beach, such a malfunction could cause the structure to strike a person or become unstable, potentially leading to the assembly tipping over and creating a safety hazard.

While the intended use of the system is to smoothly track a user to maintain consistent shade coverage, a deviation occurs if the system fails to filter environmental noise or multiple potential users. For instance, if the de-

Table 2: Bill of materials

Description	Model #	Manufacturer	Quantity	Cost @	Total
Raspberry PI 5 Kit	RPI5-128GB-8GB-MAX	Vilros	1	$194.99	$194.99
Pi Camera Module 3 (Wide)	B0310	Arducam	1	$36.99	$36.99
Nema 17 Planetary Gearset Stepper Motors	17HS19-1684S-PG51	STEPPERONLINE	2	$41.94	$83.88
TMC2209 Stepper Motor Driver	TMC2209 V1.3	BIGTREETECH Direct	1	$22.99	$22.99
9-DOF Accel/Mag/Gyro+Temp Breakout Board	LSM9DS1	Adafruit	1	$36.02	$36.02
5.5 Ft Portable Beach Umbrella	822-00-335-000-0	PICNIC TIME	1	$36.37	$36.37
12V 10Ah LiFePO4 Lithium Battery	CRIUS 10	GOLDENMATE	1	$39.99	$39.99
3D Printing	PLA 10	TechSpark	1	$1	$1
300 Pcs M3 Threaded Inserts M3 x D5 x L4	6274790	Yaocom	1	$7.99	$7.99
600 Pcs M3 Screws Assortment Kit	9741	Hapric	1	$9.99	$9.99
4Pcs 8mm Flange Coupling Connector	H420*4	Daier	1	$8.59	$8.59
5 Pcs Linear Motion Shaft 8 X 100 mm	02-001-001	Eowpower	1	$9.99	$9.99
2 Pcs R20-2RS Double Rubber Bearings	R20-2RS	XiKe	1	$12.99	$12.99
10ct 608-2RS Ball Bearings 8mm x 22mm x 7mm	SK-007	Wellgo	1	$7.99	$7.99
2 x Bevel Gear 1.5M 8mm Shaft Hole	hta230711hh000100	HARFINGTON	1	$14.66	$14.66
12.5mm 300Rpm 6 Wires CIRCUITSx2A	CP164	Comidox	1	$9.99	$9.99
480pcs M4 Screws Assortment	VG30607	VGBUY	1	$8.99	$8.99
2 x Bevel Gear 1M 8mm Shaft Hole	hta230801hh000148	HARFINGTON	1	$14.39	$14.39
					$557.71

tection system mistakenly identifies a child's movement as the primary user, it could initiate a rapid dual-axis adjustment that exceeds safe velocity limits. This scenario is particularly dangerous because the system is optimized for high-torque performance to manage the weight of the canopy. Such a failure would stem from a combination of inaccurate user detection and a lack of software-enforced motion constraints, violating the designers' responsibility to ensure safe operation in public spaces.

Harm in this situation extends beyond physical impact to the primary user or nearby beachgoers; it includes the potential for property damage and a loss of public trust in autonomous outdoor equipment. Vulnerable populations, such as children with unpredictable movement patterns or elderly individuals with limited mobility, are at the highest risk. Furthermore, individuals unfamiliar with the product may not anticipate that the structure can move automatically, placing them in the path of the motorized arm. This underscores the ethical challenge of ensuring that a device intended for accessibility and comfort does not become an unintended liability for those least equipped to respond to its movement.

To mitigate these risks, the design prioritizes several ethical values, including privacy, public health, and environmental welfare. We ensure that all camera-based user tracking is processed locally on-device via the Raspberry Pi 5 to avoid storing or transmitting personal visual data. From a public health perspective, the system actively addresses the global risk of UV exposure and heat-related illnesses by providing continuous automated shade. Additionally, the mechanical motion is constrained by software limits and hardware motor bounds to ensure operation remains safe even during sensing errors.

Finally, environmental and economic factors were central to our component selection to ensure project sustainability. The system utilizes low-cost, widely available hardware to improve affordability and scalability. We specifically chose a 12V LiFePO4 battery, which offers a longer lifespan and safer, stable output in high-heat environments compared to standard alternatives, thereby reducing environmental replacement waste. By balancing these design choices, Shade.ai aims to fulfill the ethical duty of providing a beneficial public health tool that respects user privacy and physical safety.

10 RELATED WORK

This project is related to the previous ECE capstone projects that have been dealt with autonomous drone and robot usage and other Computer Vision applications:

- Team A6: "Search and Rescue Drone" from the Fall 2024 Semester iteration
- Team A0: "Building tSAR" from the Fall 2025 Semester iteration

We, however, had to pivot away from using a drone and move to a more stationary system for our project. We also took inspiration from a company that already developed a solar-based, robotic, autonomous shading system called Shadecraft SUNFLOWER [3]. Our shade system is not solar-based, and simplifies most of the autonomous features to keep the capstone budget-friendly.

11 SUMMARY

Shade.ai presents a portable, autonomous beach umbrella system designed to eliminate the need for manual shade adjustment in dynamic coastal environments. By integrating solar position modeling, computer vision-based

user tracking, and a dual-axis motorized pan–tilt mechanism, the system works to maintain continuous shade for the user. The Raspberry Pi 5 serves as the central controller, coordinating perception, control, and Bluetooth communication while meeting strict latency and responsiveness requirements. Quantitative design targets for durability, portability, battery life, and detection latency ensure the system remains practical for real-world beach deployment.

Through structured trade studies, the team selected stepper motors with planetary gear reduction for high holding torque under wind load, a $LiFePO_4$ battery for thermal safety and extended operation, and a CV-based tracking approach to maximize usability without requiring wearable devices. Risk mitigation strategies address torque margins, power consumption, and integration complexity to improve reliability before final validation. Overall, Shade.ai demonstrates integration of embedded systems, control theory, computer vision, and mechanical design into a cohesive autonomous product that improves sun protection, accessibility, and user convenience in outdoor recreational settings.

11.1 Future Work

Future development of the Shade.ai system will focus on expanding the intelligence of the perception pipeline to support multiple users simultaneously. While the current implementation effectively utilizes the MoveNet model for single-user tracking, a multi-user detection algorithm would allow the system to calculate an optimal shade centroid that provides coverage for small groups or families. This would require more sophisticated orientation calculations and potentially a wider range of motion to ensure all detected individuals remain within the cast shadow. Refining the vision logic to prioritize specific users or lock onto a primary target would also prevent the system from becoming confused in crowded beach environments, addressing the safety considerations identified during system evaluation.

Beyond software improvements, future iterations will aim to enhance the mechanical presentation and modularity of the physical assembly. The current 3D-printed mounts and brackets, which were utilized for rapid prototyping and weight efficiency, will be refined into more aesthetically polished, professional-grade enclosures that better align with the project's IP67 durability goals. Furthermore, a significant area of development lies in creating a universal attachment interface that allows for customizable umbrella sizes and types. This modular approach would enable users to swap different canopies based on their specific needs—ranging from small personal umbrellas to larger group shelters—making the system a versatile and adaptable platform for automated sun protection. These refinements would significantly improve the product's marketability and its ability to meet diverse consumer requirements.

11.2 Lessons Learned

The development of Shade.ai underscored the necessity of deliberate hardware selection and the meticulous mapping of components to design requirements early in the process, as architectural mistakes during this initial stage prove exceptionally costly once system integration begins. One of the most significant insights gained was that hardware and mechanical validation—encompassing load management, precise alignment, and environmental constraints—requires a substantially larger time investment than software development. Real-world factors such as weight, torque, and structural stability often necessitate design iterations that are not foreseeable in simulation or software-only environments, forcing a constant recalibration between ambitious technical goals and practical feasibility.

Clear documentation of internal assumptions, interfaces, and trade-offs remains essential for both debugging and formal reporting, as internal design logic is rarely obvious to stakeholders or even other team members over the project's lifecycle. We found that effective task division and selecting a project domain where team members possess established expertise can significantly streamline the development cycle. Despite the successful integration of many subsystems, the inherent fragility of hardware was highlighted when our stepper drivers got fried immediately before demo day prevented a real-time demonstration of the umbrella's pan and tilt functionality. Ultimately, this capstone experience provided invaluable lessons in striking a balance between project ambition and mechanical robustness, reinforcing the need for redundancy in critical hardware paths.

Glossary of Acronyms

- CV - Computer Vision
- GATT – Generic Attribute Profile
- PID - Proportional, Integral, and Derivative
- RPi – Raspberry Pi

References

[1] European GNSS Agency. *GPS Accuracy: Smartphone vs. Dedicated GPS Devices*. Accessed: 2026-05-02. 2024. URL: https://www.euspa.europa.eu/newsroom/news/how-accurate-your-smartphone-gps.

[2] Huzaifa Ali. *Complementary Filter on Accelerometer and Gyroscope*. Accessed: 2026-02-23. 2025. URL: https://medium.com/@alikhuzaifaali1129/complementary-filter-on-accelerometer-and-gyroscope-b6f9aa4e49ca.

[3] Consumer Technology Association. *ShadeCraft SUNFLOWER*. Accessed: 2026-02-11. 2026. URL: https://www.ces.tech/success-stories/shadecraftsunflower/.

[4] Healthline. *Average Shoulder Width and How to Measure Yours*. Accessed: 2026-05-02. 2018. URL: https://www.healthline.com/health/average-shoulder-width.

[5] Product Design Hub. *Ergonomic Guidelines for Portable Consumer Electronics and Equipment*. Accessed: 2026-05-02. 2025. URL: https://www.productdesignhub.com/ergonomics-carrying-limits.

[6] LiTime. *LiFePO4 vs Lithium-Ion Batteries: Full Comparison*. Accessed: 2026-02-23. 2023. URL: https://www.litime.com/blogs/compare-batteries/lifepo4-vs-lithium-ion-batteries.

[7] Murcal. *Understanding IP67: A Guide to Ingress Protection Ratings*. Accessed: 2026-05-02. 2024. URL: https://www.murcal.com/blog/understanding-ip67.

[8] Mark D. Needham. *Recreation Carrying Capacity and Management at Kailua Beach Park on Oahu, Hawaii*. Tech. rep. Oregon State University, 2012. URL: https://nature.forestry.oregonstate.edu/sites/default/files/2007-2b%20HCRI%20-%20Kailua%20Beach%20Park%20-%20Final%20Project%20Report%20-%20Needham%20-%20Final.pdf.

[9] PubNub. *How Fast is Realtime? Human Perception and Technology*. Accessed: 2026-05-02. 2024. URL: https://www.pubnub.com/blog/how-fast-is-realtime-human-perception-and-technology/.

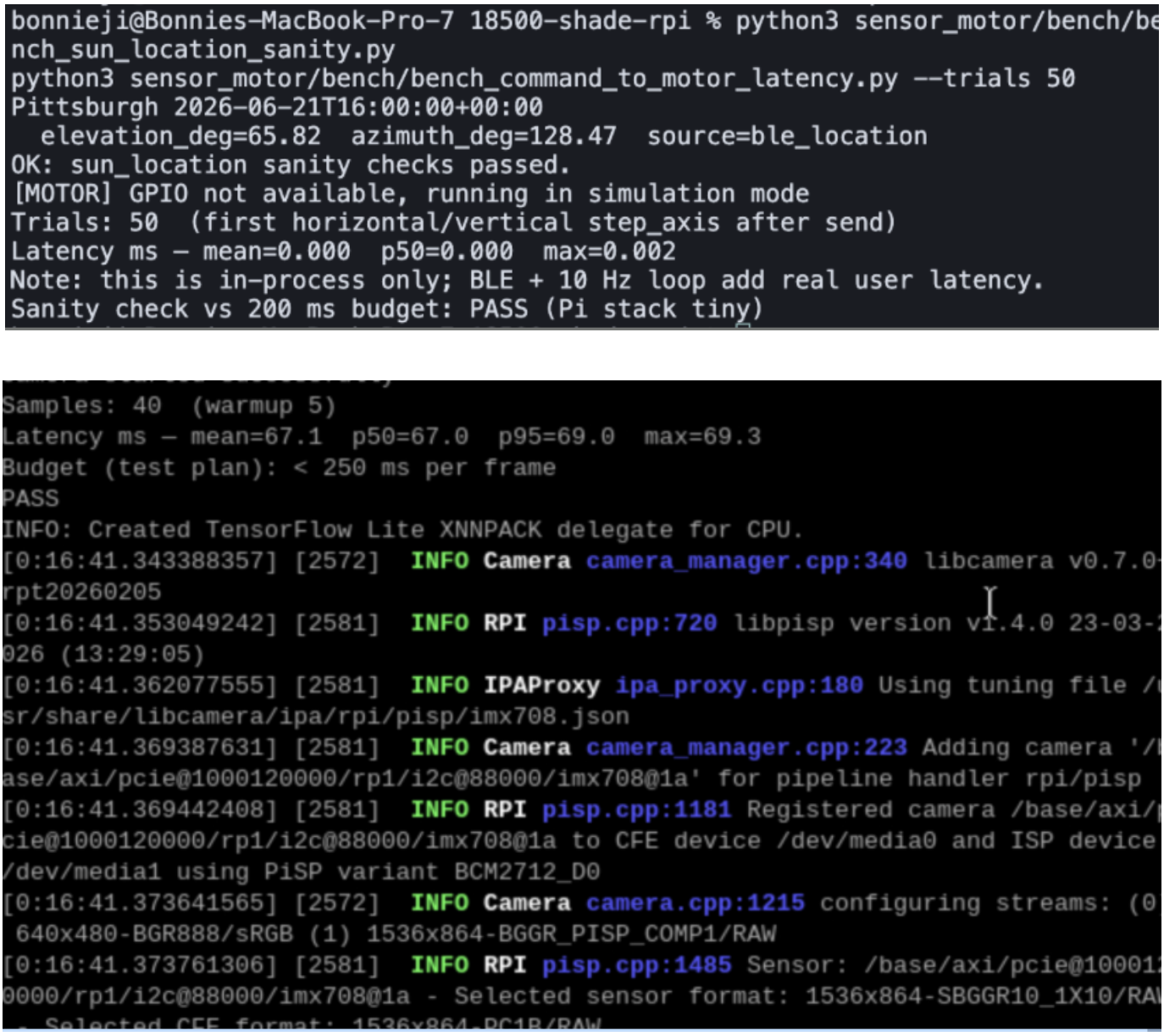


Figure 2: iOS application testing results: (top) BLE command transmission and connection validation, (bottom) system response and latency verification during manual control.

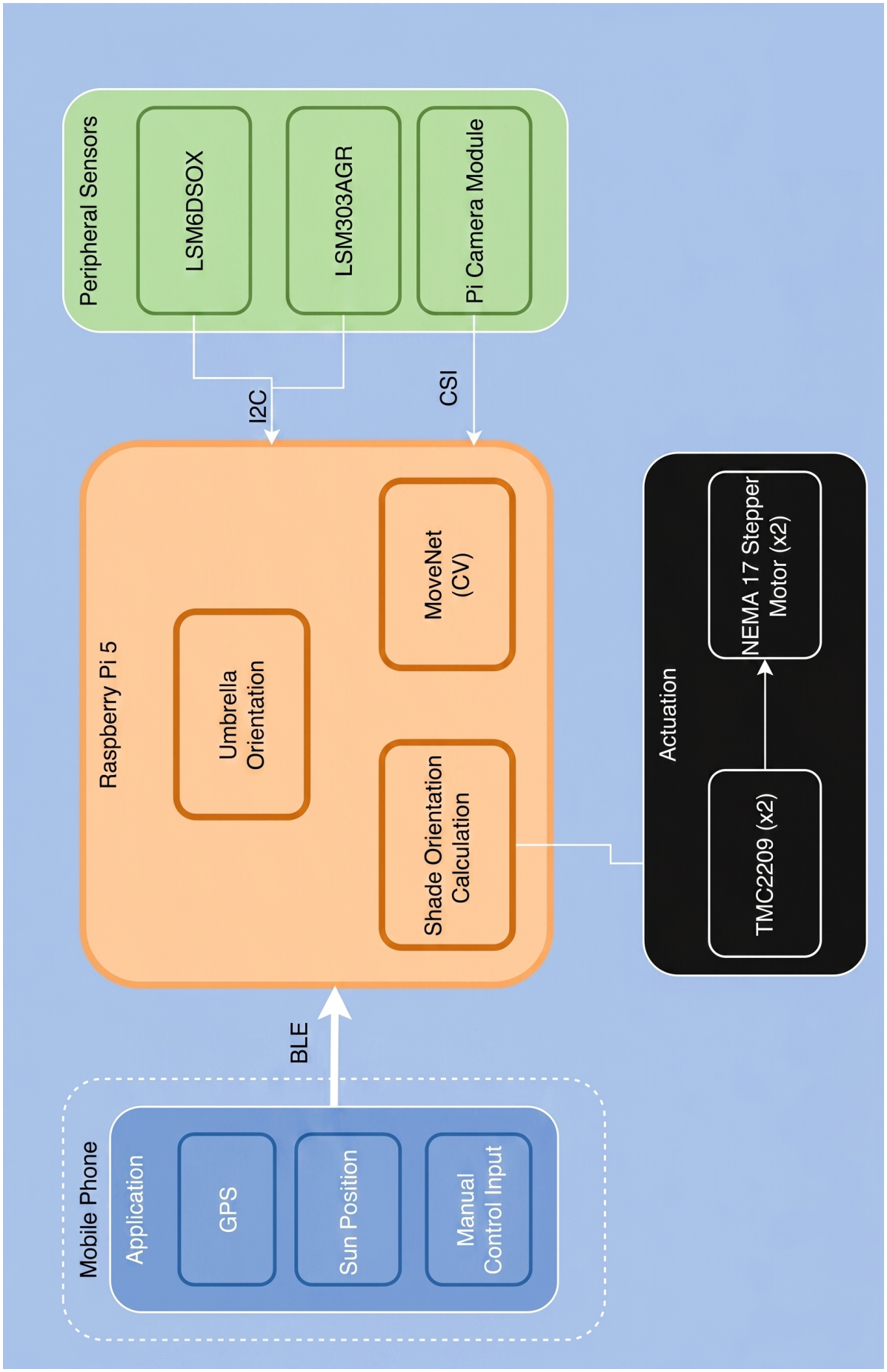


Figure 3: Full System Block Diagram

Team E5 Schedule

PROJECT TITLE	Shade.ai
Team	Ore Abolade, Bonnie Ji, Nicholas Pinon

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	PCT OF TASK COMPLETE
			January		
			WEEK 1		
			1/11 - 1/17		
1	**General Team Tasks**				
1.1	Abstract	Everyone	1/17/26	1/21/26	100%
1.2	Website Setup	Everyone	1/29/26	1/31/26	100%
1.3	Slack	Everyone	2/9	2/13	100%
1.4	Proposal Presentation	Everyone	1/28/26	2/1/26	100%
1.5	Resource Gathering	Everyone	2/1/26	2/8/26	100%
1.6	Design Presentation	Everyone	2/8/26	2/15/26	100%
1.7	Design document	Everyone	2/18/26	2/27/26	100%
1.8	Ethics Assignment	Everyone		3/11/26	100%
1.9	Interim Demo	Everyone		3/30/26	100%
1.10	Final Presentation Slide	Everyone		4/19/26	100%
1.11	Poster PDF	Everyone		4/29/26	100%
1.12	Final Video	Everyone		5/1	100%
1.12	Public Demo	Everyone		5/1	100%
1.14	Final Report	Everyone		5/2	100%
2	**Shade System Integration and Testing**				
2.1	Motor Selection & Torque Estimation	Ore + Nick	2/16/26	2/19/26	100%
2.3	Motor & Encoder Wiring & Power Setup	Everyone	3/30/26	4/6/26	100%
2.4	Full System Motion Test (No CV/User)	Everyone	4/3/26	4/7/26	100%
2.5	Sun Sensor Integration & Testing	Ore + Bonnie	4/6/26	4/30/26	80%
2.6	Full System Tracking Test (CV + Sun Sensor)	Ore + Bonnie	4/9/26	4/23/26	100%
3	**Mechanical Design & Physical Assembly**				
3.1	Arm & Joint CAD Design	Nick	2/16/26	2/23/26	100%
3.2	Component Mounting & Brackets	Nick	3/30/26	4/21	100%
3.4	Full Physical Assembly (Arm + Joints + Motors)	Everyone	3/25/26	4/21/26	100%
4	**CV/Perception**				
4.1	Environment Setup & Lighting Calibration	Ore + Bonnie	2/18/26	2/23/26	100%
4.2	User Detection Implementation	Ore + Bonnie	2/24/26	2/28/26	100%
4.3	Integration with Arm Motion (Target Positioning)	Everyone	3/30/26	4/30	60%
5	**Control & System Integration**				
5.2	Motor Communication & Encoder Feedback Setup	Everyone	3/30/26	4/6/26	100%
5.4	Full System Integration (CV + PID + Sun Sensor)	Everyone	4/9/26	4/30/26	80%
6	**App Development & UV Monitoring**				
6.1	App Backend & Motor Control API	Bonnie + Nick	3/1/26	4/7/26	100%
6.3	App Integration with Full System	Everyone	4/11/26	4/14/26	100%
6.4	User Testing & Manual Control Features	Everyone	4/14/26	4/30/26	75%

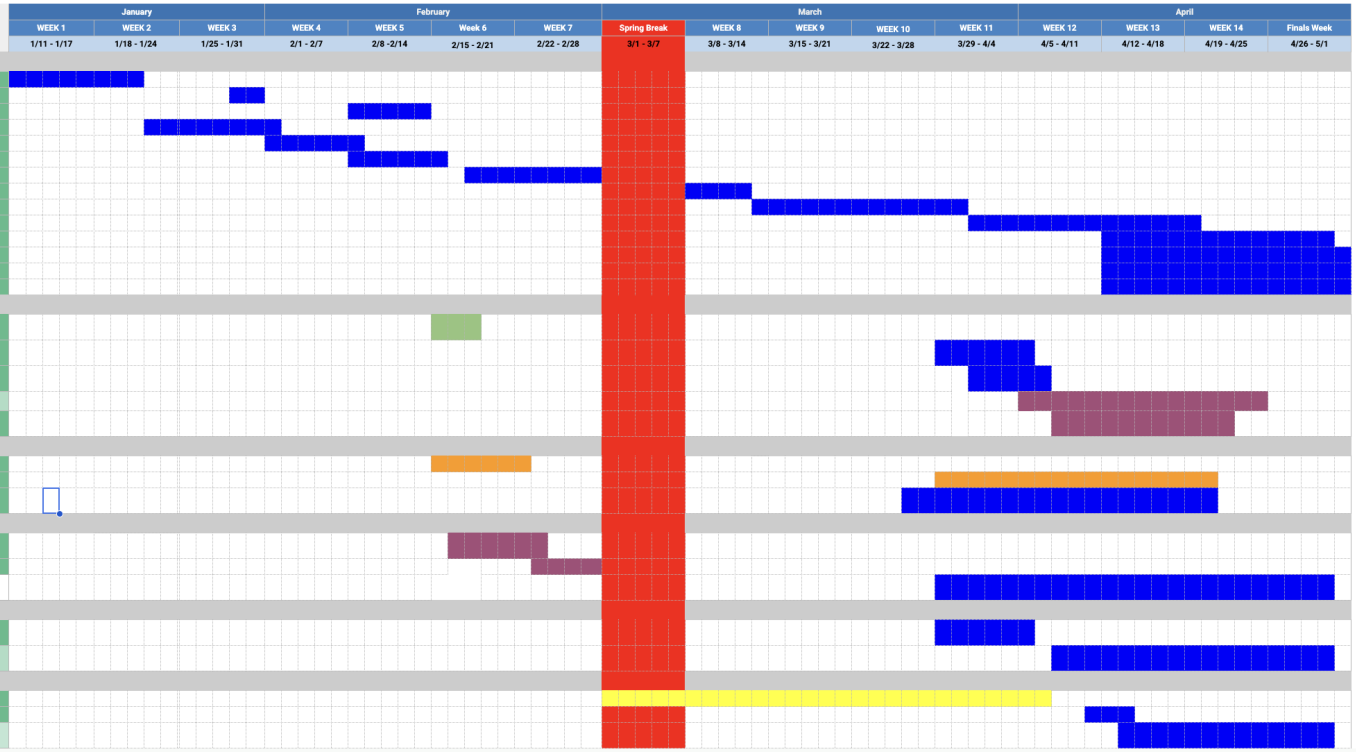


Figure 4: Gantt Chart