# Wireless Smart Sunshade Control System for Smart Homes

**Group: 09**

**Farhan Mamun (Matriculation Number: 6163382), Kazi Fahid Reza (Matriculation Number: 6163183), Abul Kalam (Matriculation Number: 6088012)**

***Keywords: Controller, Actuator, Blynk, Smart Sunshade, Servo, UDP, IOT***

***Abstract***: This project utilizes two microcontroller units to create a smart sunshade system, consisting of a central controller and an actuator. The actuator sends the LDR sensor readings of the current light intensity to the controller. Based on these readings, the controller determines how much the servo motor should move to adjust the sunshade. The target light intensity is predefined in two modes: a manual mode and an app-controlled mode via a mobile app, which allows for manual adjustments. Both units communicate wirelessly through a local access point.

***Components***: The following components were used in the implementation of the smart sunshade control system:

1. **ESP32-S2 (x2):** Microcontroller units used as the controller and actuator for wireless communication and system control.
2. **LDR Sensor (x1):** Light-dependent resistor for measuring the current light intensity in the room.
3. **Servo Motor (x1):** Used to physically adjust the sunshade position based on the control signals.
4. **Male-to-Male (M-M) Cables:** For establishing connections between the ESP32-S2 modules and other hardware components.
5. **16x2 LCD Display:** For displaying real-time information (light intensity and sunshade position).

***Challenges Identified***:

**1. Wireless Communication:** One of the key challenges encountered was ensuring reliable wireless communication between the controller and the actuator. Initially, a direct Wi-Fi connection between the two was implemented, but it proved to be unstable, with weak signal strength and frequent disconnections. To address this issue, a local access point was introduced, which significantly improved the stability and reliability of the connection.

**2. Synchronization Between Devices:** The actuator needs to receive real-time commands from the controller while frequently sending updated LDR sensor readings. For real-time communication, UDP was used to ensure swift data exchange. The system continuously monitors the light intensity, and decisions regarding sunshade adjustments are made based on the current readings and the user-defined target intensity. Keeping the devices synchronized in this fast-paced environment, especially when the user adjusts the target intensity, was a critical challenge.

**3. Sun Light Challenge in prototype:** One challenge during prototype testing was inconsistent sunlight availability, as the system relies on real-time LDR sensor readings to adjust the sunshade. This made it difficult to accurately test the system's responsiveness under varying light conditions. To overcome this, artificial light was used to simulate different scenarios, but replicating real-world lighting proved challenging for fine-tuning the system.

***System Architecture***: The system was designed with 2 different parts, they are:

**1. Controller:** The controller unit obtains live light intensity information from the actuator and enables users to adjust the desired light intensity using manual input or the mobile app, Blynk. The Blynk app offers users another way to control the system from a distance, allowing them to send commands to the controller to customize the sunshade according to their liking. Moreover, the controller also includes an LCD display to provide real-time information on the current light intensity, target intensity, and sunshade position. This enables users to check the device's status on the system itself, eliminating the need to use the mobile app. Wi-Fi and the UDP protocol enable seamless data exchange and real-time system operation through wireless communication between the controller and the actuator.

**2. Actuator:** The controller receives data on current light intensity from an LDR sensor in the actuator unit. The actuator changes the position of the sunshade as per the instructions from the controller. It operates a servo motor to manually adjust the sunshade, guaranteeing that the indoor light levels correspond to the user's specified intensity goal.

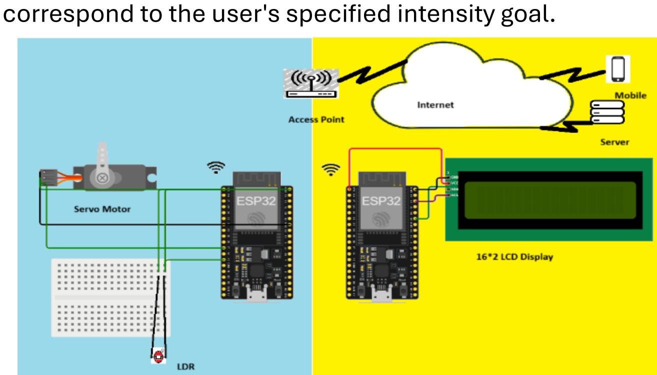
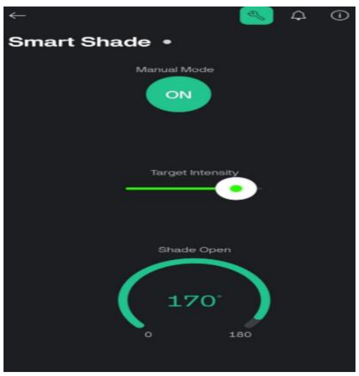



**Figure1: Block Diagram of the wireless smart sunshade.**

***Application Based solution*:** The Blynk app was integrated as a core part of the system's remote-control solution, providing users with a convenient way to manage the smart sunshade via their smartphones. The app offers two levels of access: an admin mode, where users can log in with an ID and password to control the system, and a web view (HTTP) mode that provides read-only access for monitoring purposes. This read-only mode allows users to view real-time data, such as current light intensity and sunshade position, without having control over the system. The token-based communication ensures secure interactions between the app and the controller, preventing unauthorized access. This dual-access design was implemented to enhance security and ensure that only authorized users can adjust, while anonymous users can still monitor the system without compromising its functionality.



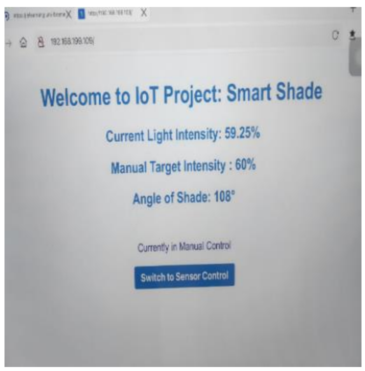
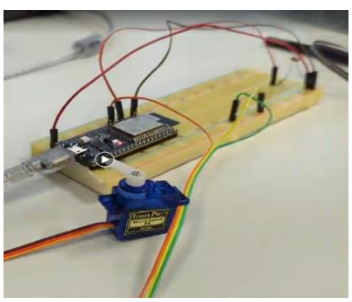




**Figure 2: Blynk app interface based.** **Figure 3: HTTP Read-only mode for data (use interface).**



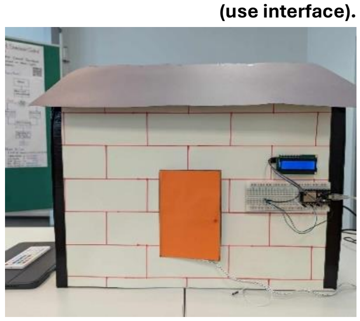


**Figure 4: Actuator Deployment and testing**

**Figure 5: Controller Deployment and testing**

The app-based communication system in Figure 2 includes a switch for manual on/off control, giving the user the option to switch between manual control through the app or automatic control. The interface includes a cursor for adjusting the desired light intensity, which in turn dictates the servo motor's rotation from 0 to 180 degrees. The choice of "shade opening" indicates the servo's current position according to the intensity chosen. In Figure 3, the web interface shows the current light intensity from the LDR sensor on the actuator and the corresponding shade angle needed to reach the desired light intensity. This interface also has a feature called "manual target intensity," which utilizes information from the Blynk app to adjust the target light. The servo motor adapts as necessary, while the LDR sensor keeps monitoring and adjusting the position until reaching the desired intensity. The setup in Figure 4 shows a servo motor and an LDR sensor working together to measure light intensity and adjust the position of the sunshade. Finally, Figure 5 demonstrates the arrangement of the controller unit, which includes an LCD screen. The screen shows live updates on the current and desired light levels, and indicates the sunshade's location, enabling users to oversee the system on-site without depending on the application.

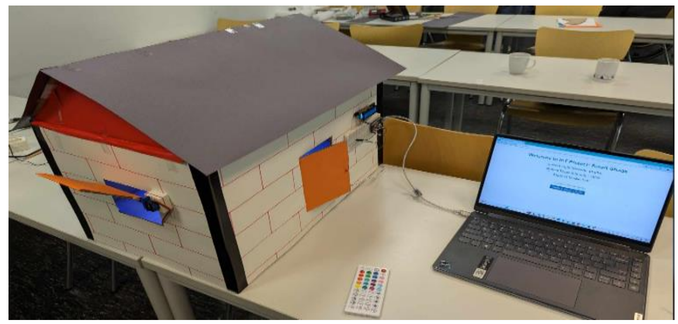


**Figure 6: Prototype implementation and testing**

***Logic Design*:** The setup functions based on an event-based structure, where actions are triggered by real-time inputs such as readings of light intensity, leading to adjustments in the position of the sunshade. The system operates like a state machine, with two main states: manual mode and sensor-based mode. When set to manual mode, the user has the option to control the sunshade's position using the Blynk app. In sensor-based mode, the sunshade is managed automatically by the system using light intensity data from the LDR sensor.

The logic is represented in the flowchart in figure 7. The system begins by obtaining the LDR value from the actuator to assess the light intensity in the room. Afterwards, the system examines the control mode to decide if it should function in manual or sensor-based mode.

- In the manual mode, the user specifies the desired intensity on the app, and the system transmits information to the actuator for accurate calibration of the servo motor, responsible for regulating the position of the shade.
- In sensor-based operation, the system automatically changes the shade angle depending on the light intensity data from the LDR sensor.

After adjusting the shade position, the system transmits the new data to the actuator for accurate servo movements. It further refreshes the current data in real-time on both the web interface and the LCD display for monitoring purposes. This process goes on, with the system consistently monitoring the LDR reading and making necessary adjustments to the shade to align with the desired light level.

UDP protocol facilitates fast communication between the actuator and the controller, allowing for immediate adjustments. Mapping values connects LDR sensor data (from 0 to 10,000) with servo angles (from 1 to 180 degrees), allowing precise manipulation of the sunshade's motion.

The system constantly compares the present light intensity with the desired light level in both modes. According to the comparison, the controller instructs the actuator to adjust the shade accordingly. Ultimately, token-based authentication depicted in Figure 8 guarantees safe interaction between the Blynk app and the controller, blocking unauthorized entry and maintaining system security

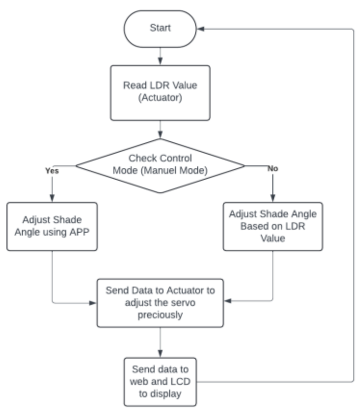



**Figure 7: Flowchart of the system.**

***Results and Performance***: The intelligent sunshade system was evaluated in different lighting situations and functioned effectively in both manual and sensor-driven settings. The actuator quickly reacted to shifts in light brightness, modifying the shade to correspond with the desired light level. The communication between the controller and actuator using UDP was found to be dependable and consistent, enabling instant modifications with little to no delay. The communication between the controller and the app was done with TCP as the controller was controlling everything and the APP was used to change the target intensity, so the reliable communication needed here. The incorporation of the Blynk application offered a seamless and intuitive experience, allowing effortless management and observation of the system through a mobile device. The LCD screen effectively displayed local monitoring data, including current and target light intensities and shade position.

***Power Consumption***: Both active and standby modes were considered when measuring the system's power consumption. Power consumption was calculated assuming the system is in active mode 60% to 40%. This method accurately evaluates how much battery is being used in a 24-hour timeframe. In actuator, one ESP32 and an LCD display was used as a component and in actuator part one ESP32, a servo motor and a LDR used. As the power consumption of LDR is very low so it was omitted in the power calculation.

- **Power Consumption of ESP32:**
  Current in active mode: 160 mA [1]
  Voltage: 5V
  Power Consumption =0.16 A×5 V =0.8W
  Current in standby mode: 20 μA [1]
  Power Consumption =0.00002 A×5 V=0.0001 W
- **Power Consumption of Servo Motor SG90:**
  maximum load (stall) current: 650 mA
  Voltage: 5V
  Power Consumption =0.65 A×5 V = 3.25W
  No load current: 10 mA
  Power Consumption =0.01 A×5 V=0.05W
- **Power Consumption of 16*2 LCD monitor:**
  Current Cunsumption: 30 mA
  Voltage: 5V
  Power Consumption =0.03 A×5 V= 0.03A×5V=0.15W

| Actuator Power Consumption | ESP32 (W) | Servo Motor (W) | Total Power Consumption |
|---|---|---|---|
| Active Mode | 0.8 | 3.25 | 4.05 |
| Standby Mode | 0.0001 | 0.05 | 0.0501 |
| Controller Power Consumption | ESP32 (W) | LCD (W) | Total Power Consumption |
| Active Mode | 0.8 | 0.15 | 0.95 |
| Standby Mode | 0.0001 | | 0.1501 |

**Table 1: Power Consumption of Controller and Actuator in different mode over an Hour**

| Actuator Power Consumption | Considering Active time Energy Consumption for 1 day | | |
|---|---|---|---|
| Active Mode | 60.00% | 50.00% | 40.00% |
| Total Power Consumption (Watt) | 58.8 | 49.2 | 39.6 |
| Controller Power Consumption | Considering Active time Energy Consumption for 1 day | | |
| Active Mode | 60.00% | 50.00% | 40.00% |
| Total Power Consumption (Watt) | 15.1 | 13.2 | 11.3 |

**Table 2: Energy Consumption of Controller and Actuator Considering active time**

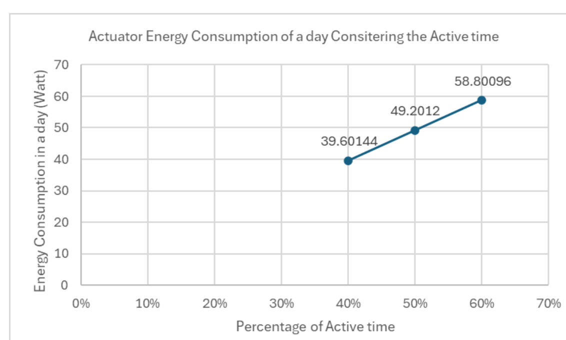



**Figure 9: Energy Consumption of Actuator considering different percentage of consumption of a day.**

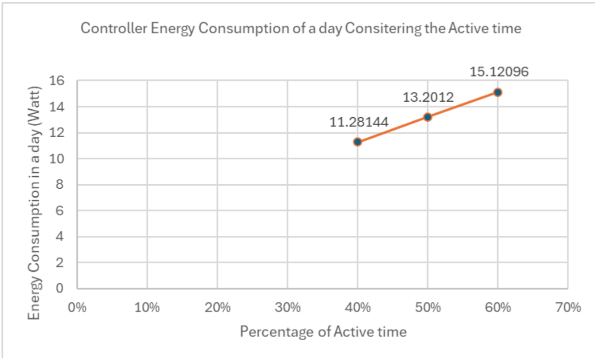



**Figure 10: Energy Consumption of Actuator considering different percentage of consumption of a day.**

***Conclusion*****:** This project effectively installed a smart sunshade control system that offers automated and manual control features. The system maintains optimal indoor lighting by modifying the shade according to real-time LDR sensor data and user-defined target light intensity. Utilizing UDP for fast communication and token-based authentication for secure entry, this system is a beneficial enhancement for any smart home. The system offers a flexible and reliable solution for automating sunshade control with strong overall performance.

***Future Scopes:***

- **Voice Control Integration:** Implement voice control capabilities using smart home assistants like Amazon Alexa or Google Home for hands-free operation of the sunshade system.
- **NTP Server Integration:** Use an NTP (Network Time Protocol) server to schedule automatic shade adjustments based on time, such as closing the shade at night and reopening it in the morning.
- **Backup Communication with ESP-NOW:** Integrate ESP-NOW as a backup communication method between the controller and actuator to improve reliability in case of Wi-Fi failures or interruptions.
- **Customizable User Settings:** Provide users with advanced customization options, allowing them to set different shade control profiles based on their preferences for various times of the day or specific scenarios (e.g., reading, relaxing).

***Reference:***

*[1] Current Consumption Measurement of Modules - ESP32-S2 - — ESP-IDF Programming Guide v5.2.3 documentation. (n.d.-a). https://docs.espressif.com/projects/esp-idf/en/stable/esp32s2/api-guides/current-consumption-measurement-modules.html*