Title : IoT-Based Dual Axis Solar Tracker Using Arduino UNO and ESP32 with Blynk Integration

Group Members Information (Group 2):

Maisha Fairooz: 21201031

Yamin Adnan: 21201356

Asif Arman Rafsan: 21201155

Mahdi Tazwar: 21301237

1.3 Abstract :

This project focuses on the development of an IoT-enabled dual-axis solar tracker system using Arduino UNO and ESP32 microcontrollers, integrated with the Blynk IoT platform for remote monitoring and control. The system aims to maximize solar energy absorption by rotating the solar panel 180° along both the X and Y axes, ensuring it constantly faces the direction of the strongest sunlight. Light Dependent Resistor (LDR) sensors are used to detect sunlight intensity, guiding the servo motors for precise alignment. The harvested solar energy is stored in a battery, which is then used to power auxiliary devices such as a light and a fan. Users can control these devices remotely through the Blynk mobile application from any location via internet connectivity. In addition, real-time data such as battery charge level, solar panel voltage, ambient temperature, and humidity are displayed on the Blynk interface, providing live environmental feedback. Communication between the ESP32 and the Blynk server is achieved through Wi-Fi, demonstrating the integration of IoT and wireless protocols in embedded systems. This project effectively combines microcontroller interfacing, sensor integration, actuator control, and IoT communication to deliver a smart energy solution. It highlights the importance of efficient energy utilization and showcases the potential of wireless technology in automating and managing renewable energy systems remotely.

2.1 Problem Statement:

Conventional solar panel systems are typically static, resulting in suboptimal energy absorption due to their fixed position relative to the sun's path. Additionally, traditional systems lack remote monitoring and control capabilities. This project addresses the need for a cost-effective, automated, and remotely manageable solution that optimizes solar energy capture and provides user-friendly control through mobile applications.

2.2 Objectives:

- To implement interfacing between Arduino UNO, ESP32, and peripheral components such as LDR sensors, temperature and humidity sensors, fan, and light.
- To utilize appropriate communication protocols (Single wire protocol and Wi-Fi) to enable data transmission between microcontrollers and the Blynk IoT platform.
- To establish real-time monitoring of environmental parameters (temperature, humidity, voltage, battery level) via sensor interfacing and display through the Blynk app.

- To configure ESP32 for Wi-Fi-based communication with the Blynk server for seamless remote control and data visualization.
- To ensure smooth operation of actuators (servo motors for solar tracking, fan, and light) through microcontroller-based interfacing and command execution via Blynk.

2.3 Significance:

This project showcases a practical application of interfacing and communication protocols in an IoT-based environment. By combining embedded systems with wireless communication and mobile applications, it demonstrates how smart energy solutions can be developed to increase the efficiency and accessibility of renewable energy systems. The project has real-world applicability in smart homes, remote locations, and sustainable infrastructure, highlighting the potential of embedded interfacing in modern automation and green technology.

3.1: Interfacing Components

- **Role:** Acts as the primary controller to process data from the LDR sensors and control the servo motors for solar panel alignment.
- **Why it's used:** Simple to program and excellent for real-time control tasks like sensor reading and servo driving.

ESP32

- **Role:** Handles IoT communication with the Blynk app via Wi-Fi. Also reads data from the DHT22 sensor and voltage sensors to send real-time updates to the app.
- **Why it's used:** It has built-in Wi-Fi and more processing power than Arduino, making it ideal for networking and data monitoring tasks.

2. Sensors

4 × LDRs (Light Dependent Resistors)

- **Role:** Detect light intensity from different directions (top, bottom, left, right).
- **Why it's used:** Helps determine the direction of maximum sunlight so the solar panel can be rotated to that position.

2 × Voltage Sensors

- **Role:** Measure voltage from:
  - The solar panel output

    - The battery output
- **Why it's used:** It lets the system monitor charging performance and the remaining battery level, which is displayed in the Blynk app.

DHT22 (Temperature & Humidity Sensor)

- **Role:** Measures ambient temperature and humidity.

- **Why it's used:** Helps monitor environmental conditions and sends data to the Blynk app.

3. Actuators

2 × Servo Motors:

- **Role:** Rotate the solar panel:
    - One on the X-axis (left-right)
    - One on the Y-axis (up-down)
- **Why it's used:** Allows precise control over the panel's angle to track the sun or light.

Relay Module:

- **Role:** Turns on/off external devices like a light and a fan.
- **Why it's used:** Provides electrical isolation and control of higher voltage/current devices through the Blynk app.

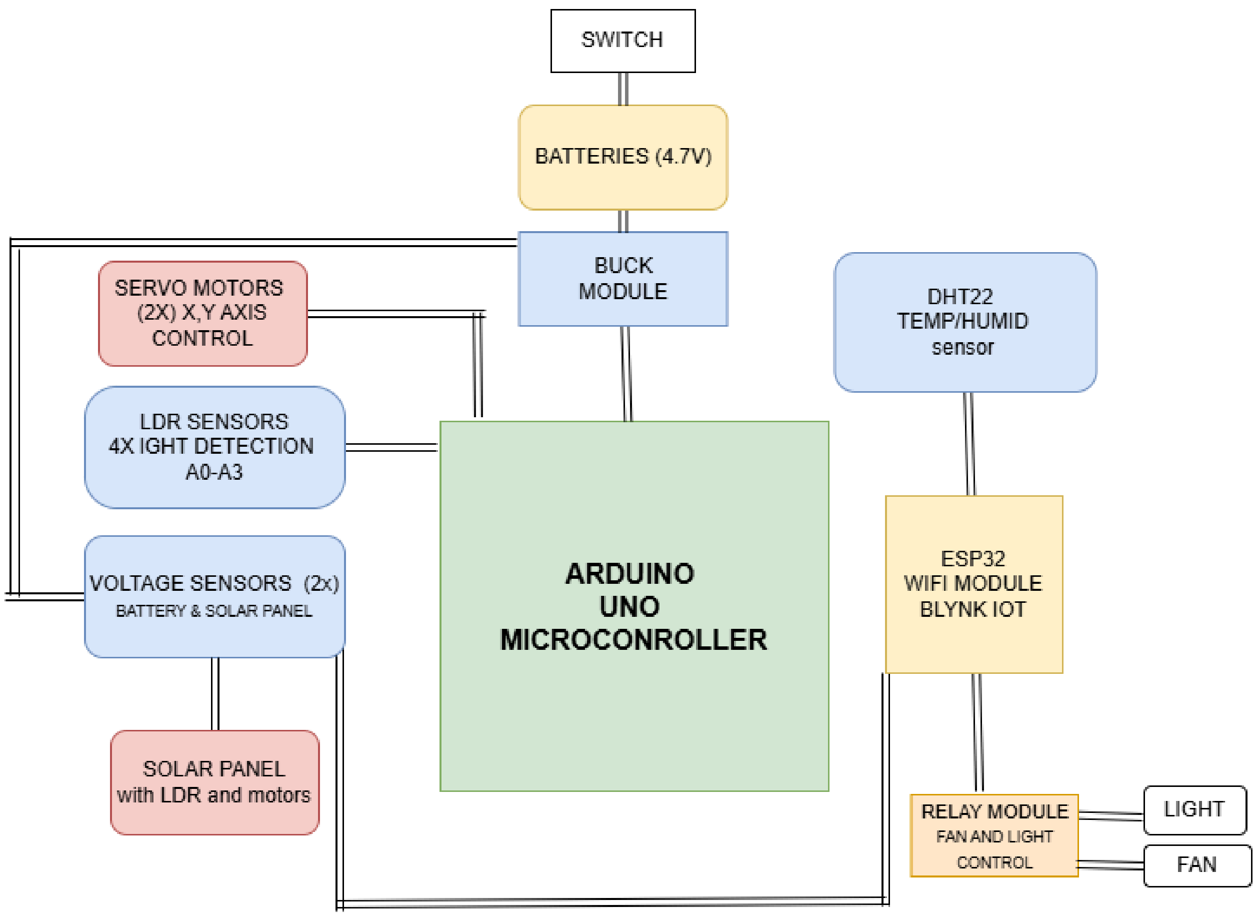
4. Power Component

- **2 × 4.2V Batteries**
- **Role:** Store the energy generated by the solar panel and supply power to the light, fan, and parts of the control system.

5. Buck Converter Module:

- **Role:** Steps down the solar panel voltage to a safe level for charging the battery or powering components.
- **Why it's used:** Protects the battery and electronics from overvoltage.

3.2 System Block Diagram:



3.3 Interfacing Challenges:

Challenges:

The Dual Axis Solar Tracker project may face several interfacing challenges. Signal integrity issues can arise from long or unshielded wires, causing voltage drops or unstable sensor readings. Analog signals from LDRs and voltage sensors are particularly vulnerable to electrical noise. Power instability is another concern, especially when servos or relays draw high current, potentially resetting the microcontrollers. Logic level mismatches between the 5V Arduino Uno and the 3.3V ESP32 can lead to communication issues or even hardware damage. Servo motors may experience jitter due to noise or an inadequate power supply. Wi-Fi connectivity issues in the ESP32 can disrupt real-time monitoring and control via the Blynk app. Additionally, timing conflicts or communication errors can occur when both Arduino and ESP32 operate simultaneously without synchronization.

Solutions:

We can use short, shielded wires and decoupling capacitors to address these challenges to maintain signal integrity and reduce voltage fluctuations.. Prevent power dips by powering servos and relays separately, and add large capacitors along with reverse protection diodes. Use logic level shifters or voltage dividers to safely interface between Arduino and ESP32. Reduce servo jitter by isolating servo power and using deadzone logic in code. Ensure a strong Wi-Fi signal for the ESP32 and implement reconnect logic in software to handle disconnections

4.1 Protocol Selection:

1. Wi-Fi protocol:

Overview: Wi-Fi, based on IEEE 802.11 standards, is a wireless protocol that supports high-speed data transfer over local networks and the internet. The ESP32's integrated Wi-Fi module connects to the Blynk cloud server, enabling remote monitoring and control through the Blynk mobile app.

2. Role: Wi-Fi transmits sensor data, including light intensity (derived from LDRs via a voltage sensor), battery charge (via a voltage sensor), and temperature and humidity (from the DHT22), to the Blynk app. It also receives commands to control the relay module for operating the light and fan.

Single-Wire (One-Wire) protocol:

Overview: The single-wire protocol, a specific type of serial communication, uses a single data line for bidirectional communication between a master device (ESP32) and a slave sensor (DHT22). It is designed for low-bandwidth data transfer, transmitting data sequentially as timed digital pulses.

Role: The single-wire protocol connects the DHT22 sensor to the ESP32, facilitating the collection of temperature and humidity data for environmental monitoring and display in the Blynk app.

The four LDRs and two voltage sensors (one for LDR data, one for battery charge) are interfaced via analog pins on the ESP32 and Arduino Uno, using their built-in analog-to-digital converters (ADC) rather than a communication protocol. The servo motors (HL525 and MG90S) are controlled via PWM signals, and the relay module is toggled via digital GPIO pins on the ESP32. These components are managed through direct pin control, not communication protocols.

4.2 Protocol Justification:

Wi-Fi:

Justification: Wi-Fi was chosen for its long-range and high-speed capabilities, critical for enabling remote monitoring and control via the Blynk app. The ESP32's Wi-Fi module supports 802.11 b/g/n standards, providing data rates up to 150 Mbps, which is more than sufficient for transmitting sensor data (light intensity, battery charge, temperature, and humidity) and receiving control commands for the relay module with low latency. Wi-Fi's range (up to 100 meters in open environments, depending on the environment) ensures reliable connectivity to a local router, facilitating internet-based access from any location. The protocol's compatibility with the Blynk platform, which relies on cloud-based communication, makes it the optimal choice. Alternatives like Bluetooth were considered but rejected due to their limited range (~10 meters) and lack of native support for cloud-based applications like Blynk.

Considerations: Wi-Fi consumes more power than low-energy protocols, but the solar-powered system, with two 4.7V batteries and a buck converter, provides sufficient energy to sustain the ESP32's operation.

Single-Wire (One-Wire):

Justification: The single-wire protocol was selected for the DHT22 temperature and humidity sensor due to its compatibility and minimal hardware requirements. The DHT22 uses a single data pin to transmit digital temperature and humidity readings, with a data rate sufficient for periodic updates (every 2 seconds, per the sensor's specification). This protocol is ideal for monitoring environmental conditions without occupying multiple GPIO pins on the ESP32, enabling direct integration with the Blynk app. The single-wire protocol's short-range capability (up to a few meters) is suitable for the compact layout of the solar tracker's electronics. Alternatives like I2C were considered, but the DHT22's native single-wire interface made it the most straightforward choice, avoiding additional hardware or protocol conversion.

Considerations: The single-wire protocol requires precise timing for reliable data transfer, but standard ESP32 libraries (e.g., DHT library) handle this efficiently, ensuring accurate readings.

5.1 Methodology:

Simplified Pseudocode for IoT Solar Monitoring (ESP32)

Protocol Configuration

SET Blynk credentials: TEMPLATE_ID, TEMPLATE_NAME, AUTH_TOKEN
SET Wi-Fi credentials: ssid = "student", pass = "iotstudent"
SET Serial baud rate = 9600
SET DHT22 on pin 25
SET Blynk virtual pins: V0 (temp), V1 (humidity), V3 (solar volt), V4 (charge), V5 (relay1), V6 (relay2)
SET GPIO: voltageSensor (34), voltageSensor1 (35), relay (26, output), relay1 (27, output)

Main File

```
FUNCTION setup():
   INIT Serial at 9600
   INIT DHT22
   SET relays as OUTPUT, HIGH (off)
   CONNECT Wi-Fi with ssid, pass
   CONNECT Blynk with AUTH_TOKEN
END FUNCTION

FUNCTION loop():
   RUN Blynk
   READ battery voltage, CALCULATE charge (map 7.10V-8.40V to 0-100%)
   READ solar voltage
   READ temperature, humidity from DHT22
   SEND to Blynk: V0 (temp), V1 (humidity), V3 (solar volt), V4 (charge)
   DELAY 1000 ms
END FUNCTION

FUNCTION Blynk_V5_callback(value):
   SET relay1 to LOW if value = 1, else HIGH
END FUNCTION

FUNCTION Blynk_V6_callback(value):
   SET relay2 to LOW if value = 1, else HIGH
END FUNCTION
```

Simplified Pseudocode for Solar Tracker (Arduino Uno)

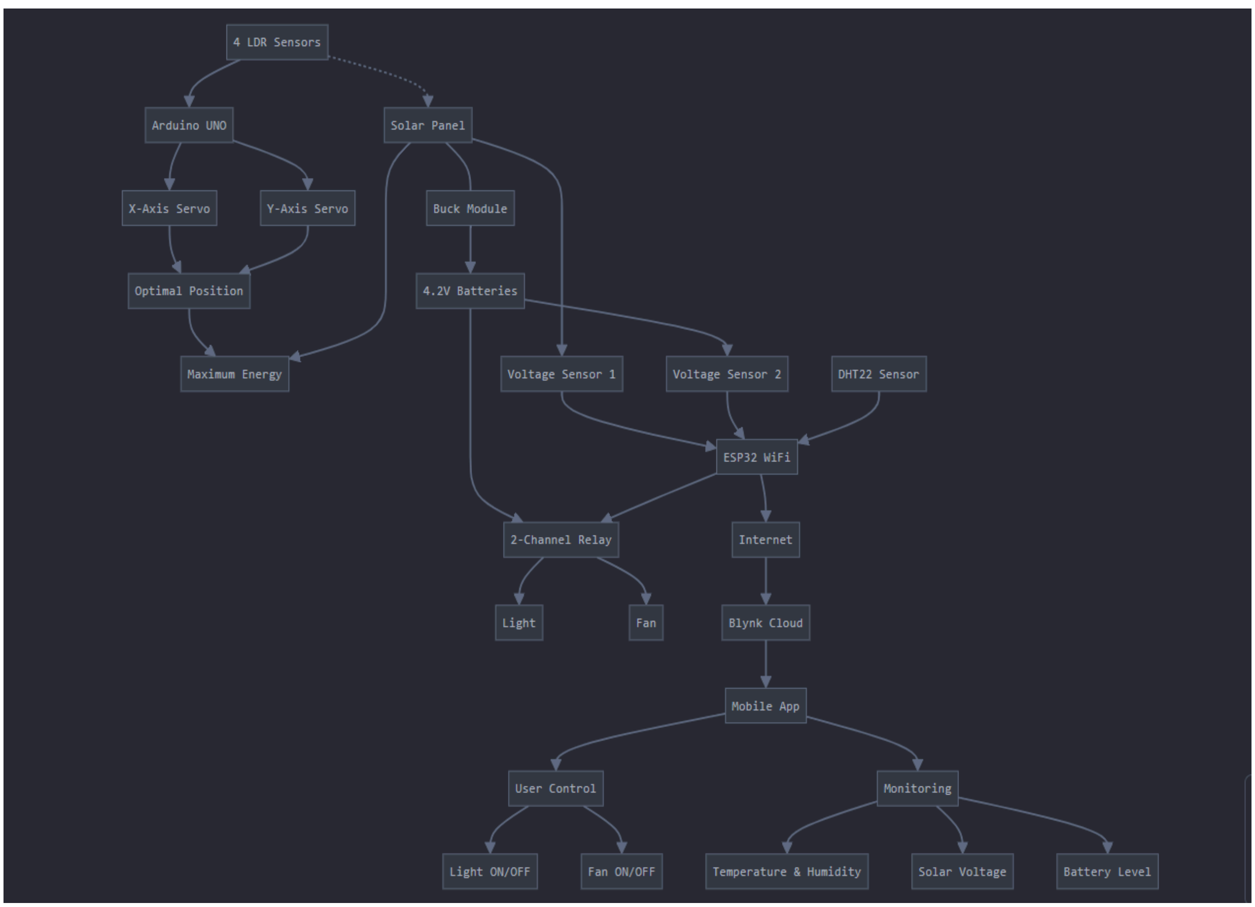
Protocol Configuration

SET Serial baud rate = 9600
SET PWM Timer 1: ICR1 = 40000, OCR1A = 3000, OCR1B = 3600
SET GPIO: pin 9 (PWM output), pin 10 (PWM output), A0-A3 (LDR inputs)

Main File
FUNCTION setup():
INIT Serial at 9600
SET pins 9, 10 as OUTPUT
CONFIGURE Timer 1 for PWM
END FUNCTION

FUNCTION loop():
READ LDRs: topleft (A0), topright (A1), downleft (A2), downright (A3)
ADJUST OCR1A based on topleft vs topright, downleft vs downright
ADJUST OCR1B based on topright vs downright, topleft vs downleft
CLAMP OCR1A between 2000 and 4000
DELAY 1 ms per adjustment
END FUNCTION

Flowchart:



5.2 Expected Outcomes

Functionality:

- Tracks sunlight accurately with LDRs and servo motors to optimize energy capture.
- Monitors solar voltage, battery charge, temperature, and humidity in real-time via Blynk app.
- Enables low-latency remote control of light and fan from anywhere with internet.

Reliability:

- Uses robust protocols: Wi-Fi/Blynk for IoT, PWM for servos, DHT for sensors.
- ESP32 ensures stable cloud connectivity; Arduino provides precise motor control.
- Buck module maintains 5V supply, but Wi-Fi signal or servo precision may affect reliability.

Potential Issues:

- Wi-Fi instability may disrupt Blynk; mitigable with extenders.
- Servo accuracy may falter with heavy panels; high-torque motors needed for scaling.

6.1 Future Improvements:

Interfacing Reliability:

- Add UART/I2C for Arduino-ESP32 data sharing (e.g., LDR data to Blynk).
- Include local buttons for relay control during Wi-Fi outages.
- Auto-calibrate LDRs/voltage sensors for environmental changes.

Protocol Efficiency:

- Use MQTT for low-bandwidth communication.
- Send differential data to reduce network load.
- Add RTC for scheduled load operations.

6.2 Applications:

- **Off-Grid Systems:** Powers remote homes with IoT energy management.
- **Smart Agriculture:** Runs irrigation/ventilation, monitors climate.
- **Portable Generators:** Support camping/emergencies with remote control.
- **Education:** Teaches IoT, solar energy, and embedded systems.
- **Microgrids:** Scales for community energy with IoT monitoring.

References :

1. Al-Sheikh, M. A. (2025). IoT-enabled dual-axis solar tracking system using ESP32 and Blynk for real-time monitoring and energy optimization. Jupiter: Publikasi Ilmu Keteknikan Industri, Teknik Elektro dan Informatika, 3(1), 187–204.https://www.researchgate.net/publication/388106720_IoT-Enabled_Dual-Axis_Solar_Tracking_System_Using_ESP32_and_Blynk_for_Real-Time_Monitoring_and_Energy_Optimization?fbclid=IwY2xjawKLP4hleHRuA2FlbQIxMABicmlkETE5V1pjbXpWZkQxYVo2bXlsAR6wiS_7rQNv50i4082IycRLuyinwb6iIDbMnS3DD4Ubiu98dpaQdCT3vlf09A_aem_urDv2s0Lk-T6NrL43HH1HA

2. Al-Sheikh, M. A. (2025). Advanced dual-axis solar tracking system with IoT-driven real-time monitoring for optimized efficiency. Jupiter: Publikasi Ilmu Keteknikan Industri, Teknik Elektro dan Informatika, 3(2), 210–225. https://www.researchgate.net/publication/390153623_Advanced_Dual-Axis_Solar_Tracking_System_with_IoT-Driven_Real-Time_Monitoring_for_Optimized_Efficiency?fbclid=IwY2xjawKLQBJleHRuA2FlbQIxMABicmlkETE5V1pjbXpWZkQxYVo2bXlsAR7ACXF9NzMTF2qUWB5gfQe1M84ftAUmMHQ5qpahgnwSQr25-MBqWfHUOunVSA_aem_Mat8N5Zzq4M0mxdaFdBM9A

3. Achbi, M. S., Rouabah, B., Benarabi, B., Mahboub, M. A., & Kechida, S. (2024). IoT-based monitoring of a dual-axis solar tracking system. Przegląd Elektrotechniczny, 100(7), 62–66.https://pe.org.pl/articles/2024/7/14.pdf?fbclid=IwY2xjawKLQGZleHRuA2FlbQIxMABicmlkETE5V1pjbXpWZkQxYVo2bXlsAR6npXU70n3_CwGCgjlam9Px0tyIs41yj5HnoGL0H0HieN68S7s-ZvJDrPO6cw_aem_ixge8pRPY0s2lvtl4ZC1XQ