1. LED Light Source

- Use **high-lumen, low-wattage LED chips** (e.g., 200 lumens per watt) to maximize brightness and efficiency.
- Opt for a colour temperature of **4000–5000K** for optimal visibility and reduced glare.
- Integrate a high-quality aluminium heat sink for heat dissipation.

2. Solar Panel

- Use **monocrystalline solar panels** for maximum efficiency (~20–22%).
- Size the solar panel to generate at least **50% more energy** than required to compensate for inefficiencies and cloudy days. For an 18-hour runtime, calculate the LED's total wattage and size the panel accordingly.
- Include a **tilt-adjustable mount** with a servo motor controlled by a **basic PLC** to track the sun's position.

3. Battery System

- Use a **LiFePO4 (Lithium Iron Phosphate)** battery for durability, safety, and deep discharge capability.
- Battery capacity should be at least **1.5 times the daily requirement** to ensure reliability (e.g., for a 60W light: 60W × 18h = 1080Wh, so use a ~1500Wh battery).
- Include a **Battery Management System (BMS)** to protect against overcharging and deep discharging.

4. Charge Controller

- Use an **MPPT (Maximum Power Point Tracking)** controller for efficient energy conversion from the solar panel to the battery.
- Ensure the controller is compatible with the solar panel and battery voltage.

5. Cooling and Heat Dissipation

- For LEDs: Use **extruded aluminium heat sinks** with natural convection fins.
- For the unit: Design with **ventilation holes** or passive cooling techniques to avoid additional costs.

6. Automation for Solar Tracking

- Use a low-cost **micro-PLC** or **Arduino-based system** with a **light-dependent resistor (LDR)** or a small solar sensor array to track sunlight.
- Implement **2-axis tracking** for optimal efficiency.

- Use a small **12V servo motor** or actuator for movement.

7. Pole Design

- Use lightweight yet sturdy materials such as **galvanized steel** or **aluminium**.
- Include an **anti-corrosion coating** for outdoor use.
- Ensure the design allows easy access to the battery and electronics for maintenance.

8. Cost Optimization

- Prioritize **local suppliers** for materials and components to reduce costs.
- Offer modular designs for municipalities to choose between basic and premium features.

Final System Specifications for One Street Lamp

1. **LED Light**: 60W, high-efficiency, 4000–5000K.
2. **Battery**: 120Ah, 12V LiFePO4 battery.
3. **Solar Panel**: 300W monocrystalline panel.
4. **Charge Controller**: 12V, 20A MPPT controller.
5. **Solar Tracking System**: Cost-effective PLC/Arduino with a servo motor.

Processes to Build In-House

1. **Solar Panel Manufacturing:**
    - **Machines Used:**
        - Laminator for Solar Panels
        - Stringer for Solar Cells
        - Module Testing Equipment
    - **Tasks Done In-House:**
        - Connect solar cells into series/parallel configurations using the **Stringer**.
        - Laminate the cells with tempered glass, EVA, and a back sheet using the **Laminator**.
        - Test solar panels for efficiency using the **Module Testing Equipment**.
2. **Battery Pack Assembly:**
    - **Machines Used:**
        - Spot Welding Machine for Battery Packs
        - BMS Integration Tools

- **Tasks Done In-House:**
    - Assemble battery packs by spot-welding lithium-ion cells.
    - Integrate the Battery Management System (BMS) into battery packs.
    - Conduct charge-discharge cycle testing for quality assurance.

3. **Lighting Unit Assembly:**
    - **Machines Used (Shared with Soft Starter Line):**
        - Pick-and-Place Machine
        - Reflow Oven
        - Solder Paste Printer
    - **Tasks Done In-House:**
        - Assemble LED PCBs by mounting LEDs (diodes) and drivers using the SMT line.
        - Attach PCBs to heat sinks using thermal interface material.
        - Test LED modules for illumination, power efficiency, and thermal performance.
4. **Pole Assembly and Integration:**
    - **Machines Used:**
        - Manual workstations for welding, drilling, and painting.
    - **Tasks Done In-House:**
        - Weld mounting brackets to poles for solar panels and light fixtures.
        - Drill holes for wiring and fixtures.
        - Paint poles for corrosion resistance and aesthetics.
        - Assemble the pole with the solar panel, battery, and lighting unit.
5. **Solar Tracking System Integration:**
    - **Tasks Done In-House:**
        - Install cost-effective PLCs with motors for auto-adjusting solar panels.
        - Program and test solar tracking functionality.

Processes to Outsource

1. **Raw Materials for Poles:**
    - **Outsource Galvanized Steel:** Procure 12 m pre-cut galvanized steel sections to avoid heavy machining in-house.

2. **Solar Cells:**
    - **Outsource Solar Cells:** Purchase ready-made solar cells to streamline assembly.
3. **LEDs (Diodes):**
    - **Outsource Bulk LEDs:** Procure high-efficiency diodes from manufacturers like Cree or Osram to assemble onto PCBs.
4. **Glass for Solar Panels:**
    - **Outsource Tempered Glass:** Purchase low-iron tempered glass for use in solar panels.
5. **Enclosures for Light Fixtures:**
    - **Outsource Aluminium/Plastic Enclosures:** Procure ready-made enclosures for LED modules to reduce fabrication costs.
6. **BMS Units:**
    - **Outsource BMS Boards:** Purchase pre-designed Battery Management System (BMS) boards for easy integration.

Summary

By keeping the **solar panel manufacturing**, **battery pack assembly**, **LED module assembly**, and **pole integration** in-house, we retain control over critical quality aspects. Outsourcing **raw materials** (like galvanized steel and tempered glass), **solar cells**, **LEDs**, and **BMS boards** will minimize initial capital investment and reduce production complexity.

Comprehensive Cost Analysis for One Solar-Powered Street Lamp Assembly

1. Material Costs

A. Solar Panel

- **Solar Cells**: ₹20/cell × 72 cells = ₹1,440
- **Tempered Glass (Low-Iron)**: ₹500/panel
- **EVA Sheet and Back Sheet**: ₹300/panel
- **Aluminium Frame**: ₹400/panel
- **Junction Box and Cables**: ₹200/panel
- **Solar Panel Total**: ₹2,840

B. Battery Pack (Lithium-Ion)

- **Lithium-Ion Cells**: ₹150/cell × 40 cells = ₹6,000
- **BMS (10s/24V)**: ₹800/unit

- **Enclosure (Metal/Plastic)**: ₹500/unit
- **Wiring and Connectors**: ₹200/unit
- **Battery Pack Total**: ₹7,500

C. LED Lighting Unit

- **LEDs (Diodes)**: ₹50/diode × 50 LEDs = ₹2,500
- **LED Driver Circuit**: ₹300/unit
- **PCB (Fabrication + SMT Assembly)**: ₹400/unit
- **Heat Sink (Aluminium)**: ₹600/unit
- **Enclosure for LED Unit**: ₹800/unit
- **LED Unit Total**: ₹4,600

D. Pole and Mounting Structure

- **Galvanized Steel Pole (12 m)**: ₹6,000/unit
- **Painting and Corrosion Protection**: ₹800/unit
- **Mounting Brackets for Solar Panel/LED Unit**: ₹600/unit
- **Pole Total**: ₹7,400

E. Solar Tracking System

- **PLC (Low-Cost Unit)**: ₹1,500/unit
- **Motor with Gears**: ₹1,000/unit
- **Wiring, Sensors, and Connectors**: ₹500/unit
- **Solar Tracking System Total**: ₹3,000

2. Manufacturing Costs

In-House Processes

1. **Solar Panel Assembly**:
    - Cost of laminating, stringing, and assembling: ₹400/panel
2. **Battery Pack Assembly**:
    - Spot welding, testing, and BMS integration: ₹600/unit
3. **Lighting Unit Assembly**:
    - PCB assembly, heat sink mounting, and testing: ₹500/unit
4. **Pole Assembly**:
    - Welding, painting, and bracket installation: ₹500/unit

5. **Solar Tracking System Integration**:
    - PLC programming and motor assembly: ₹400/unit
6. **Quality Control and Testing**:
    - ₹500/unit

- **Manufacturing Subtotal**: ₹2,900

3. Labour Costs

- Skilled Labour (Technicians): ₹200/hour × 8 hours = ₹1,600/unit
- Unskilled Labour (Assembly/Material Handling): ₹100/hour × 8 hours = ₹800/unit
- **Labour Subtotal**: ₹2,400

4. Overheads and Miscellaneous Costs

- **Factory Overheads (Electricity, Maintenance, etc.)**: ₹500/unit
- **Packaging**: ₹200/unit
- **Transportation to Client Site**: ₹1,000/unit
- **Miscellaneous (Spare Parts, Scrap, etc.)**: ₹400/unit
- **Overheads Subtotal**: ₹2,100

5. Total Cost Price of One Street Lamp

Category	**Cost (₹)**
Material Costs	₹25,340
Manufacturing Costs	₹2,900
Labour Costs	₹2,400
Overheads and Miscellaneous	₹2,100
Total Cost Price	**₹32,740**

6. Profit Margin and Selling Price

To ensure profitability and cover unforeseen expenses, assume a **profit margin of 30%**.

- **Profit**: ₹32,740 × 30% = ₹9,822
- **Selling Price**: ₹32,740 + ₹9,822 = **₹42,562**

Tentative Selling Price

Round up for simplicity and market competitiveness:

- **Final Selling Price per Unit**: **₹43,000**

Summary

- **Cost Price**: ₹32,740 per unit
- **Profit Margin**: ₹9,822 (30%)
- **Selling Price**: ₹43,000 per unit

Market Analysis

Based on the market analysis, solar street lights in India are available across a wide price range, typically from ₹3,999 to ₹32,999, depending on specifications such as wattage, battery capacity, and additional features.

Our proposed selling price of ₹43,000 per unit is higher than the typical market range. However, it's important to consider the specific features and quality of our product. If our solar street lamp offers superior specifications, such as higher wattage, advanced battery technology, enhanced durability, or additional functionalities like a solar tracking system, the higher price point may be justified.

To ensure competitiveness, consider the following steps:

1. **Benchmarking**: Compare our product's specifications and features with those of competitors in the same price bracket.
2. **Value Proposition**: Clearly communicate the unique benefits and superior features of our product to potential customers.
3. **Cost Optimization**: Explore opportunities to reduce production costs without compromising quality, which could allow for a more competitive pricing strategy.

By aligning our product's features and pricing with market expectations and effectively communicating its value, we can position our solar street lamp competitively in the market.

What should we do to make the prices more competitive?

1. Optimize Production Costs

- **Bulk Procurement of Components:**
  - Negotiate bulk discounts with suppliers for LEDs, batteries, solar panels, and other components.
  - Establish long-term contracts with reliable suppliers to lock in lower prices.
- **Localize Supply Chain:**
  - Source components locally to reduce import duties, shipping costs, and lead times.
- **In-House Manufacturing:**
  - Increase in-house production for certain components, like battery assembly or PCB manufacturing, if it reduces costs over outsourcing.

- **Improve Material Usage:**
    - Reduce wastage during metal cutting, painting, and assembly processes using lean manufacturing techniques.
- **Energy-Efficient Factory Operations:**
    - Install renewable energy sources like solar panels on our factory to reduce operational costs.

2. Streamline Design

- **Simplify the Design:**
    - Reassess our design for unnecessary complexities that increase production time and cost.
- **Use Standardized Components:**
    - Opt for standardized or off-the-shelf parts (e.g., solar panel sizes, mounting brackets) instead of custom-built components.
- **Modular Construction:**
    - Design the street lamps for modular assembly, which speeds up production and reduces labour costs.

3. Increase Production Efficiency

- **Automate Processes:**
    - Use pick-and-place machines and robotic arms for repetitive tasks to reduce labour costs.
- **Skilled Labour Training:**
    - Train employees to increase productivity and reduce errors, especially for tasks like assembly and soldering.
- **Optimize Production Workflow:**
    - Organize the production line for maximum efficiency, using tools like Kanban boards or ERP systems to manage inventory and production schedules.

4. Reduce Transportation and Distribution Costs

- **Distribute Regionally:**
    - Set up regional warehouses or production hubs closer to key markets to save on transportation costs.
- **Optimize Packaging:**
    - Use lightweight, compact packaging materials to minimize shipping costs while ensuring product safety.

5. Invest in Technology and Automation

- **Solar Panel Efficiency:**
  - Source high-efficiency, low-cost solar panels to maximize energy output and reduce panel size.
- **Battery Innovation:**
  - Explore alternative battery technologies like sodium-ion or enhanced lithium-ion to cut costs.
- **Cheaper Control Systems:**
  - Use affordable PLCs or microcontrollers like ESP32 instead of expensive controllers for sun-tracking systems.

6. Explore Economies of Scale

- **Mass Production:**
  - Increase the scale of production to reduce per-unit fixed costs.
- **Expand the Customer Base:**
  - Target larger contracts with municipalities or industrial projects for consistent demand.

7. Reassess Margins and Pricing

- **Flexible Pricing:**
  - Offer different models at varied price points, e.g., economy, standard, and premium, to cater to a broader market.
- **Subscription Model:**
  - For industrial customers, consider offering financing options or subscription-based models for installation and maintenance.

8. Government Incentives and Grants

- **Leverage Subsidies:**
  - Apply for government schemes or subsidies on renewable energy products.
- **Tax Benefits:**
  - Ensure compliance with renewable energy incentives to lower production taxes.

9. Focus on Marketing and Branding

- **Highlight Cost Savings:**
  - Emphasize long-term savings in energy costs for customers to justify the higher upfront price.

- **Target Key Markets:**
    - Focus on customers who prioritize quality, such as municipalities, highway developers, or industrial zones.

Key Actions to Reduce Price

By applying these strategies, we could aim to bring down the cost price per unit from ₹43,000 to ₹35,000–₹38,000. This could allow us to sell our solar street lamp at ₹40,000–₹42,000, making it competitive while retaining reasonable margins.

Using the TMS320F28027 for controlling the sun tracking system

The **TMS320F28027** microcontroller (MCU) from Texas Instruments is compatible for sun tracking and other control-related tasks in our solar-powered street lamp system. Here's an analysis of its suitability and how we can use it for the sun-tracking application:

Features of the TMS320F28027 Relevant to Sun Tracking

1. **High-Performance Processing:**
    - Based on the **C2000 Piccolo family**, this MCU is designed for real-time control applications. It can handle the computations needed for sun-tracking algorithms efficiently.
    - Its fast response time is ideal for tracking systems that require real-time adjustments.
2. **PWM Generation:**
    - The MCU has **PWM channels** that can be used to control the stepper or DC motors driving the solar panel adjustment mechanism.
    - You can implement precise control over motor speed and positioning.
3. **Analog-to-Digital Converter (ADC):**
    - The built-in **12-bit ADC** can measure signals from light sensors (e.g., photodiodes or LDRs) to detect the sun's position.
    - Multiple ADC channels allow monitoring of multiple sensors for fine-grained sun-tracking accuracy.
4. **Low Power Consumption:**
    - The TMS320F28027 is power-efficient, making it suitable for a solar-powered application where energy optimization is critical.
5. **I²C and UART Interfaces:**
    - These communication interfaces can be used to connect external peripherals like additional sensors or a user interface.
6. **Compact Size and Cost-Effectiveness:**

- The MCU is compact and cost-effective, aligning well with our goal of keeping production costs low.

How It Fits in the Sun-Tracking System

1. Inputs to the MCU:

- **Light Sensors (LDRs/Photodiodes):**
  - Place multiple light sensors at different angles to detect the sun's position.
  - The sensors' signals are fed to the ADC of the MCU to determine the sun's relative position.
- **Position Feedback:**
  - Use rotary encoders or potentiometers for feedback on the panel's position.

2. Processing:

- **Sun-Tracking Algorithm:**
  - Implement an algorithm such as a simple differential measurement (comparing light intensity across sensors) or a more advanced predictive tracking algorithm based on solar position formulas.
- **Control Logic:**
  - The MCU processes sensor data and calculates the required movement to align the panel with the sun.

3. Outputs from the MCU:

- **Motor Control Signals:**
  - Generate PWM signals for the motor driver circuit controlling stepper motors or DC motors.
- **Status Feedback:**
  - Communicate with an optional display or transmit status data to a remote monitoring system via UART or I²C.

Compatibility and Challenges

1. **Motor Control:**
   - The TMS320F28027 supports motor control applications, so it can easily handle sun-tracking motor control with appropriate drivers (e.g., L298N for DC motors or A4988 for stepper motors).
2. **Sensor Interface:**
   - It supports interfacing with light sensors. You'll need to calibrate the sensors for consistent performance in varying weather conditions.
3. **Processing Power:**

- While the TMS320F28027 is sufficient for sun tracking, if our system expands (e.g., integrating advanced predictive algorithms or IoT capabilities), consider more advanced MCUs from the same family.

4. **Programming:**
   - The MCU supports programming in **C/C++**, and you can use **Code Composer Studio (CCS)** for development. Ensure you have experience with TI's toolchain.

Steps to Implement Sun Tracking with TMS320F28027

1. **Hardware Setup:**
   - Connect light sensors to ADC pins.
   - Interface motors with motor drivers controlled by the MCU's PWM outputs.
   - Provide power to the MCU via a regulated supply from the solar system.
2. **Software Development:**
   - Write a program to:
     - Read ADC values from sensors.
     - Compare light intensities and compute directional adjustment.
     - Control motor positions via PWM signals.
   - Debug and test using TI's **Code Composer Studio (CCS)**.
3. **Optimization:**
   - Optimize the system for minimal motor movements to save power.
   - Include a "sleep mode" for periods when tracking is unnecessary (e.g., at night).

Conclusion

The TMS320F28027 is a suitable choice for sun tracking in our solar-powered street lamp system. It offers sufficient computational power, ADC capabilities, and PWM control for real-time adjustments while keeping costs in check. By reusing this MCU across our product lines, we streamline development, reduce inventory complexity, and save costs. Just ensure that motor drivers, sensor calibration, and software development are aligned with our design goals.

System 1: Automatic Switching for Day/Night Functionality

To ensure that the solar-powered street lamp automatically turns on when the ambient light is low (e.g., as the night falls or on a very cloudy day) and turns off when the ambient light is adequate (e.g., during sunrise), we can implement the following system:

1. Ambient Light Sensing:

- **Light Sensor (e.g., LDR or Photodiode):**
    - Place a light-dependent resistor (LDR) or photodiode on the lamp's pole to sense the ambient light levels.
    - As the ambient light decreases (e.g., at dusk or when clouds obscure the sunlight), the sensor's resistance will change, triggering the system to switch on the light.
    - When the ambient light increases (e.g., at sunrise or when sunlight returns), the sensor detects the higher intensity, and the light should be turned off.

2. Microcontroller Integration:

- **TMS320F28027 or Similar MCU:**
    - The MCU continuously monitors the light sensor input using an ADC pin.
    - When the light intensity falls below a preset threshold (for example, 5–10 lux for dusk), the MCU sends a signal to turn on the lamp.
    - When the light intensity exceeds the threshold (e.g., 50–100 lux for sunrise or brighter daylight), the MCU signals to turn off the lamp.

3. Switching Mechanism:

- **Relay or Solid-State Relay (SSR):**
    - The relay or SSR will control the power to the LED light. The relay is triggered by the MCU output to switch on or off the light.
- **Backup Mechanism:**
    - Optionally, include a delay timer to prevent false triggering during temporary changes in light conditions (e.g., momentary cloud cover).

4. Calibration:

- **Threshold Setting:**
    - The light threshold value (e.g., 10 lux) should be calibrated based on the region's typical ambient light levels. This calibration ensures that the light switches on and off at the appropriate times, regardless of seasonal or geographical differences.

System 2: Sun Tracking System Working Principle

The sun-tracking system adjusts the position of the solar panel to maximize energy absorption by aligning it directly with the sun's rays throughout the day. Here's how the system works and how the angle is set:

1. Basic Concept of Solar Tracking:

- Solar panels generate maximum power when they are directly facing the sun.

- The **solar panel** adjusts its **azimuth (horizontal angle)** and **elevation (vertical angle)** based on the sun's movement.
- The system tracks the sun's path using sensors, motors, and a microcontroller to move the panel accordingly.

2. Sensors:

- **Light Dependent Resistors (LDRs) or Photodiodes:**
  - Use a set of **LDRs** placed on the panel at different positions (e.g., left, right, up, down).
  - These sensors measure the intensity of light coming from the sun at different angles.
  - The **sun's position** is inferred from comparing the light intensities from the sensors, where the sensor receiving the most light is assumed to be facing the sun.

3. Sun Tracking Algorithm:

- The sun's position changes throughout the day due to the Earth's rotation, so the solar panel needs to adjust to these variations.
- **Two-axis tracking:**
  - **Horizontal (Azimuth) Tracking:** Controls the panel's rotation along the East-West axis.
  - **Vertical (Elevation) Tracking:** Controls the panel's tilt along the North-South axis.
- The goal is to align the solar panel to the position where the **sunlight is maximized**, and this is done by moving the panel in small increments towards the direction of stronger light intensity.
- The algorithm calculates the optimal tilt and rotation angles at different times of the day using the sensor data.

4. Working Principle of the Sun-Tracking System:

- **Morning:**
  - As the sun rises in the east, the sensors detect the increasing light on the east side of the panel.
  - The **panel is adjusted** to track the sun towards the east (or morning sun angle).
- **Midday:**
  - At solar noon, the sun is highest in the sky (elevation is at its peak).
  - The panel adjusts to maintain optimal tilt and azimuth alignment.

- **Evening:**
    - As the sun sets in the west, the sensors detect the light decreasing on the east side and increasing on the west side.
    - The panel adjusts to face the setting sun, ensuring maximum sunlight absorption.

5. Microcontroller and Motor Control:

- **TMS320F28027 MCU:**
    - The **TMS320F28027** reads the values from the sensors and calculates the necessary adjustments.
    - It generates PWM signals to control the **stepper motors or DC motors** that adjust the angle of the solar panel.
- The microcontroller checks the light intensity from each sensor, compares them, and uses this data to adjust the position of the solar panel using stepper motors or DC motors connected to motor drivers.
- The system needs to move the panel in **small increments** throughout the day to optimize sunlight capture.

6. Motor Control:

- **DC Motors or Stepper Motors:**
    - **DC motors** are simpler but can be less precise, requiring a more complex control system.
    - **Stepper motors** provide better precision and are ideal for applications that need incremental movement to position the solar panel accurately.
- The **PWM control** generated by the MCU adjusts the motors' speed and direction, tilting and rotating the panel to maintain optimal sunlight orientation.

7. Motor and Position Feedback:

- **Rotary Encoders/Potentiometers:**
    - These components can be used to provide feedback on the solar panel's angle, ensuring it is correctly positioned.
    - This feedback is used by the MCU to fine-tune the motor control and prevent overcorrection or mechanical damage.

Summary of Sun Tracking System Operation:

1. **Sensors** detect the sun’s position through light intensity differences.
2. The **TMS320F28027 microcontroller** calculates the necessary movement and controls **motors** to adjust the panel's azimuth and elevation.

3. The system continually **tracks the sun's position** by comparing sensor readings and adjusting the panel's angle for maximum energy efficiency.
4. **Automatic adjustments** occur based on real-time data, ensuring that the panel faces the sun throughout the day, maximizing solar energy capture.

Algorithm for Automatic Switching System (Day/Night Control)

This system uses a light sensor to detect the ambient light and automatically switches the light on or off based on the threshold value.

Steps:

1. Continuously read the ambient light level from the light sensor (LDR/Photodiode).
2. If the light intensity falls below a certain threshold (for example, 10 lux), switch the lamp on.
3. If the light intensity exceeds the threshold (e.g., 50 lux), switch the lamp off.
4. The system should use PWM or relay control to operate the street lamp.

Algorithm for Sun Tracking System

This system uses a set of sensors (LDRs) to track the sun's position and adjust the angle of the solar panel accordingly.

Steps:

1. Continuously read the light intensity from the LDRs (placed on different positions of the solar panel).
2. Compare the light intensities from each sensor to determine the direction of the sun.
3. Adjust the angle of the solar panel (both azimuth and elevation) using motors (stepper/DC motors).
4. Use feedback from encoders (if available) to ensure accurate positioning of the panel.

Pseudocode

Automatic Switching System

```
Initialize ADC for Light Sensor

Initialize Relay or PWM for Lamp Control

Set LIGHT_THRESHOLD = 50  // Lux value for switching


While (true):

    LIGHT_INTENSITY = Read_ADC(Light_Sensor)

    If LIGHT_INTENSITY < LIGHT_THRESHOLD:
```

```
        Turn_ON_Lamp()
    Else:
        Turn_OFF_Lamp()
    Delay(1000)  // Delay for 1 second
End While
```

Sun Tracking System

```
Initialize ADC for LDRs (Light Sensors)
Initialize Motors (Stepper/DC Motors for azimuth and
elevation)
Initialize Encoder Feedback (if using encoders)
Initialize PWM for motor control

While (true):
    LDR_1 = Read_ADC(LDR1)  // Left sensor
    LDR_2 = Read_ADC(LDR2)  // Right sensor
    LDR_3 = Read_ADC(LDR3)  // Top sensor
    LDR_4 = Read_ADC(LDR4)  // Bottom sensor

    If LDR_1 > LDR_2:
        Rotate_Left_Motor()   // Rotate panel towards the left
    Else If LDR_2 > LDR_1:
        Rotate_Right_Motor()  // Rotate panel towards the
right

    If LDR_3 > LDR_4:
        Tilt_Up_Motor()  // Rotate panel upwards
    Else If LDR_4 > LDR_3:
        Tilt_Down_Motor()  // Rotate panel downwards

    Adjust_Azimuth()    // Fine-tune horizontal movement
```

```
    Adjust_Elevation()  // Fine-tune vertical movement


    Delay(1000)  // Delay for 1 second before next sensor
reading
End While
```

C Code for TMS320F28027 in Code Composer Studio (CCS)

Day/Night Control Code

```
#include "F2802x_Device.h"  // TMS320F28027 header file


#define LIGHT_THRESHOLD 50     // Lux value for switching


void main(void)
{
    // Initialization
    InitSysCtrl();            // Initialize the system control
(clock, etc.)
    InitGpio();               // Initialize GPIO pins
    InitAdc();                // Initialize ADC for light
sensor reading
    InitEPwm();               // Initialize PWM for lamp
control (if PWM is used)


    // Initialize Relay or Lamp Control GPIO pin
    EALLOW;
    GpioCtrlRegs.GPAMUX1.bit.GPIO0 = 0;  // Configure GPIO0
for relay control
    GpioCtrlRegs.GPADIR.bit.GPIO0 = 1;   // Set as output
    EDIS;


    uint16_t light_intensity = 0;
```

```
    while(1)
    {
        light_intensity = Read_ADC(LIGHT_SENSOR_CHANNEL);  //
Read light intensity

        if(light_intensity < LIGHT_THRESHOLD)  // Light level
below threshold
        {
            Turn_ON_Lamp();
        }
        else  // Light level above threshold
        {
            Turn_OFF_Lamp();
        }

        DELAY_US(1000);  // 1-second delay
    }
}

uint16_t Read_ADC(uint16_t channel)
{
    // Start ADC conversion and read the result
    AdcRegs.ADCTRL2.bit.SOC_SEQ1 = 1;  // Start conversion for
the specified channel
    while (AdcRegs.ADCST.bit.INT_SEQ1 == 0);  // Wait until
conversion is complete
    return AdcRegs.ADCRESULT1;  // Return the ADC result
}

void Turn_ON_Lamp()
{
```

```
    GpioDataRegs.GPASET.bit.GPIO0 = 1;  // Turn on the lamp
(relay activation)
}


void Turn_OFF_Lamp()
{
    GpioDataRegs.GPACLEAR.bit.GPIO0 = 1;  // Turn off the lamp
(relay deactivation)
}
```

Sun Tracking System Code

```
#include "F2802x_Device.h"  // TMS320F28027 header file


#define LDR_THRESHOLD 100   // Threshold for sun position
adjustment


void main(void)
{
    // Initialization
    InitSysCtrl();             // Initialize system control
(clock, etc.)
    InitGpio();                // Initialize GPIO for motor
control and sensors
    InitAdc();                 // Initialize ADC for LDR sensor
reading
    InitEPwm();                // Initialize PWM for motor
control


    uint16_t ldr1, ldr2, ldr3, ldr4;  // LDR readings for sun
tracking


    while(1)
    {
```

```
// Read the LDRs to detect the sun position
ldr1 = Read_ADC(LDR1_CHANNEL);  // Left LDR
ldr2 = Read_ADC(LDR2_CHANNEL);  // Right LDR
ldr3 = Read_ADC(LDR3_CHANNEL);  // Top LDR
ldr4 = Read_ADC(LDR4_CHANNEL);  // Bottom LDR

// Adjust azimuth (left/right)
if (ldr1 > ldr2)
{
    Rotate_Left_Motor();
}
else if (ldr2 > ldr1)
{
    Rotate_Right_Motor();
}

// Adjust elevation (up/down)
if (ldr3 > ldr4)
{
    Tilt_Up_Motor();
}
else if (ldr4 > ldr3)
{
    Tilt_Down_Motor();
}

// Fine-tune the positions
Adjust_Azimuth();
Adjust_Elevation();

DELAY_US(1000);  // 1-second delay
```

```
    }
}

uint16_t Read_ADC(uint16_t channel)
{
    // Start ADC conversion and read the result
    AdcRegs.ADCTRL2.bit.SOC_SEQ1 = 1;  // Start conversion for the specified channel
    while (AdcRegs.ADCST.bit.INT_SEQ1 == 0);  // Wait until conversion is complete
    return AdcRegs.ADCRESULT1;  // Return the ADC result
}

void Rotate_Left_Motor()
{
    // Control left motor to rotate solar panel left
    GpioDataRegs.GPASET.bit.GPIO1 = 1;
    GpioDataRegs.GPACLEAR.bit.GPIO2 = 1;
}

void Rotate_Right_Motor()
{
    // Control right motor to rotate solar panel right
    GpioDataRegs.GPACLEAR.bit.GPIO1 = 1;
    GpioDataRegs.GPASET.bit.GPIO2 = 1;
}

void Tilt_Up_Motor()
{
    // Control up motor to tilt panel up
    GpioDataRegs.GPASET.bit.GPIO3 = 1;
```

```
    GpioDataRegs.GPACLEAR.bit.GPIO4 = 1;
}

void Tilt_Down_Motor()
{
    // Control down motor to tilt panel down
    GpioDataRegs.GPACLEAR.bit.GPIO3 = 1;
    GpioDataRegs.GPASET.bit.GPIO4 = 1;
}

void Adjust_Azimuth()
{
    // Fine-tune horizontal motor for precise sun alignment
    // Implement fine-tuning logic if needed
}

void Adjust_Elevation()
{
    // Fine-tune vertical motor for precise sun alignment
    // Implement fine-tuning logic if needed
}
```

Explanation of Code:

1. **Day/Night Control (Automatic Switching)**:
    - The code uses the ADC to read the light sensor value and compares it with the threshold. The relay or PWM controls the street light based on the ambient light level.
2. **Sun Tracking**:
    - The code reads the LDR values for each side of the solar panel and compares them to adjust the panel's position.
    - Motors control the azimuth and elevation of the panel based on the LDR readings.
    - Fine-tuning logic can be added for more precision in motor control.

Next Steps:

1. **Hardware Setup**:
   - Ensure that ADCs, LDRs, motors, and feedback mechanisms (encoders, if needed) are connected correctly to the TMS320F28027.
   - Configure GPIO pins as needed for relay control, motor control, and sensor inputs.
2. **Testing**:
   - Test both systems (day/night control and sun tracking) separately and then together to ensure seamless integration.