Motor Design For Sun-Tracking VLC

Key Goals/ Aims:
1. Precise solar tracking to maximize incident light.
2. Low power consumption, since it's embedded in a light-powered system.
3. Reliable orientation feedback/control (via sensors).
4. Compact, cost-effective implementation (suitable for microcontrollers like Apollo4 or STM MCU models).

Defining Sun-Tracking Requirements:
- **Degrees of Freedom:** The tracker will move on two axes, including azimuth and elevation.
- **Update Rate:** The device will adjust per a polling signal/ requirement of data transmission.
- **Accuracy:** The device will require tight alignment, where the device is reasonably able to siphon the most out of the sun.
- **Environment:** Figuring out the exposure of the device will be necessary for any device implementation outside in pure testing condition, but for the prototype the best approach will be testing inside under normal laboratory conditions.

Motor Design:

<u>General Motor Types:</u>

Motor Type	Advantages	Disadvantages	Notes
Servo	Easy control, accurate positioning	Limited rotation (usually 180°)	Good for single-axis trackers
Stepper	Good accuracy, no feedback needed	Power-hungry, can lose sync	Use with driver + current limiting
DC Motor	Simple, wide rotation range	Needs encoder or feedback for accuracy	Less ideal unless PID loop implemented
Geared DC	High torque at low RPM	Slower, might need limit switches	Useful for heavier solar panels

After considering all these motors my application seems to be based on dual axis sun tracking, light-load, occasional stepping, and precision needed.

Generally for the task that is being looked at either a “Servo” or “Stepper” Motor approach would be ideal especially for low-power tracking tasks. Though we know our design will be idle and not being powered to ensure low usage of power.

Stepper Motors here generally have a certain level of power draw even when they idle, known as “holding current”. This is because stepper motor’s coils are energised to maintain the rotor’s position, even when not actively moving — **This is not a characteristic that we want.** But we can disable this characteristic as we don’t need to maintain torque in this instance by just disabling the coils after each move using a driver setting.

We want a system that can utilise near 360° of rotation in varied steps. — **A stepper motor best fits this use case, but so does a servo motor if the design is ensured to be an open loop.** That is as both motors can achieve a full 360 degrees of continuous rotation, one that can step through all 360° and one that is continuous. A stepper can hold a certain degree of rotation, but draw current at the same time, called “holding/idling”. A servo can also be a part of this discussion, but the problem is that with servo motors the degree of rotation varies only from 180° to 270° [Which might be still applicable to this application].

<u>Exploring the use of either Steppers or Servos;</u>

Feature	Servo Motor	Stepper Motor
Typical Voltage Range	4.8V to 6V (common), 5V logic control	5V to 12V (logic & motor often separate)
Typical Current Draw	~100–500 mA idle, up to 1–2 A under load	~100–600 mA per phase (can spike >1A)
Torque	Moderate (0.5–2.5 kg·cm for SG90, MG996R)	Higher for NEMA 17 or geared steppers
Precision	~1° (controlled by PWM)	~0.9° to 7.5° per step with 400 to 48 steps usually (microstepping improves)
Ease of Control	Very easy (just PWM signal)	Moderate (requires step + dir control, driver IC)
Power Efficiency	High at idle (uses power only when moving)	Constant power draw (even when stationary)
Feedback	Built-in position feedback (potentiometer)	Open-loop (no position feedback unless encoder added)
Rotation Range	Limited (typically 180°, some up to 270° or continuous)	Unlimited (ideal for full sun sweep)
Cost (typical)	Very cheap (SG90 ~$2, MG996R ~$5–10)	Cheap (28BYJ-48 ~$2), NEMA 17 ($10–30)
Microcontroller Load	Low	Medium (needs pulse timing or driver lib)

Startup Complexity	Low (1 PWM pin)	Moderate (driver + logic pins)
Best Use	1-axis light-load tracking	2-axis, higher load or precision tracking

Considering my options either using a stepper over a servo seems more fit for the application at hand, especially with low power requirements. Motor types such as the **"28BYJ-48"** or the **"NEMA 17"** with current limiting would work best.

- Stepper motors support dual-axis rotation, making it ensure that 180-270° is not a limitation.
- Better angular control/precision, as the motor can step in increments as small as ~0.088° (with microstepping).
- Holding torque isn't as necessary in your case, since the design will idle between adjustments, but steppers are still stable in set positions.
- No continuous power draw required, as the stepper can have its coils disabled after each move using driver settings.

Option One: 28BYJ-48 (5V) + ULN2003 Driver

- Inexpensive
- Geared
- Works well with light loads
- Good for testing and POC.
- **Not so strong and might struggle with high friction or mechanical backlash**

Option Two: NEMA 17 (5V-12V) + A4988 Driver

- More precise and powerful
- Works with microstepping (1/16th step = 0.1125° with 200 Steps/rev)
- Can be current-limited to reduce power draw when idle
- Very overkill for light loads.

→ Proceeding with option One for POC work.

Expecting to purchase two 28BYJ-48, and an A4988 external driver to handle the power requirements of the stepper motor.
Power Considerations: Each Motor draws ~250mA max, but only when stepping.
While at Idle the drivers can be turned off via the A4988 to avoid any current draw.
The entire system will either be powered by a USB or by a benchtop power supply (when under lab testing conditions).

Stepper Motor - 28BYJ-48 x 2
Power Controller - A4988/ ULN2003

Sensing for Feedback:

This section will mainly focus on the approach to tracking the sun and knowing its location:

Some options are;

- LDR Array (Light dependent resistors): Placing four-ish LDRs in a cross pattern with dividers. If light falls unevenly, adjust the motor until balance is restored.
- PhotoDiodes: More faster, but more sensitive than LDRs.
- GPS + Real-time clock: Using astronomical data to calculate/determine the sun's position based on time/location. [Too high power and computation heavy]

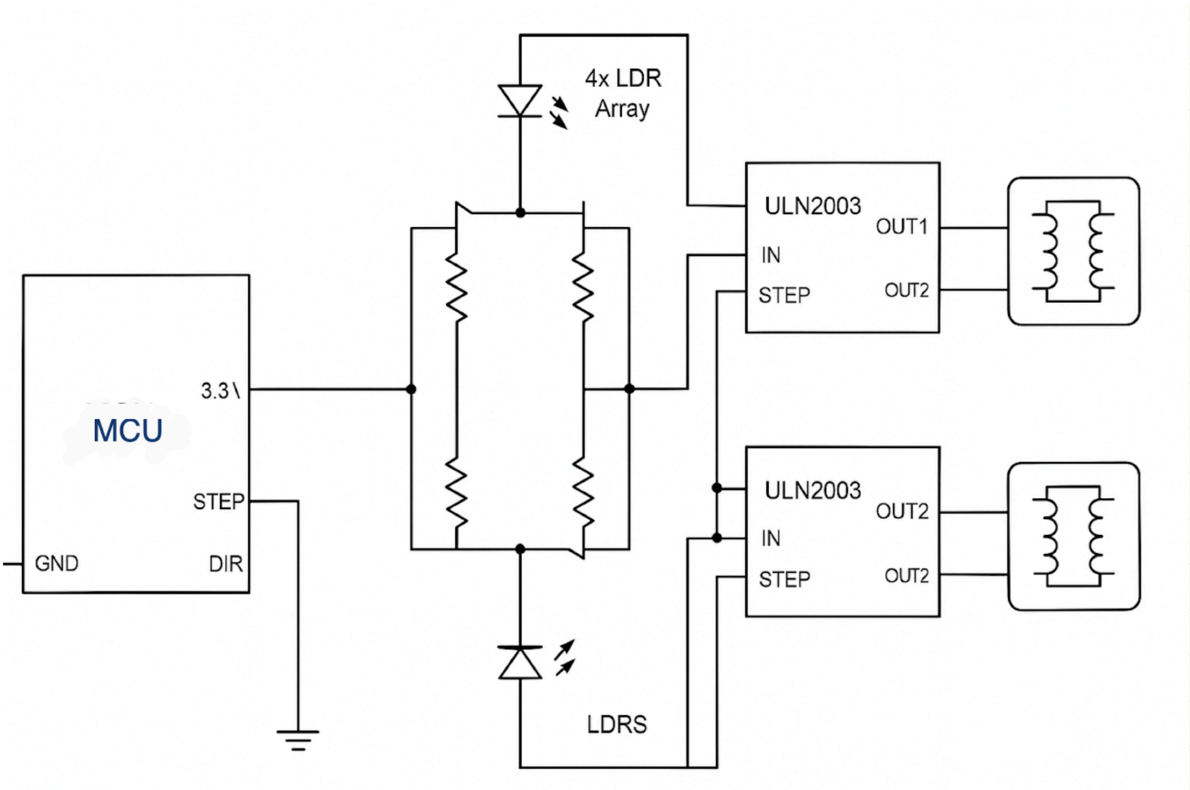
Control Algorithm:

Using the microcontroller either of a STM32 Series or Apollo4 series;

- Read sensor Feedback (ADC for LDRs)
- Calculate Required Position
- Drive Motor VIA PWM using a driver such as a

For LDR control we could use Bang-Bang control: If the LDR difference > threshold, move the motor in a direction of “light”.

PID Control: For continuous smooth adjustment, especially with DC motors + encoders.



Full system Diagram: