

A Dual-Axis Smart Solar Tracking System Using Labview and Microcontroller Interfacing

Ayush Satyam¹, Shuktika Mallaya², Riddhi Tambi³

Instrumentation Engineering Vellore Institute of Technology Vellore, India

Abstract—The orientation of a photovoltaic panel with respect to incident sunlight affects the amount of light received by its surface. A fixed panel cannot remain optimally aligned as the apparent position of the Sun changes. This paper describes the design of a low-cost, dual-axis solar-tracking control prototype based on four light-dependent resistors (LDRs), an Arduino Nano, and a virtual instrument developed in NI LabVIEW. The LDRs are arranged in opposing pairs to sense relative illumination in the horizontal and vertical directions. Their voltage-divider outputs are acquired through the Arduino's analog inputs and communicated to LabVIEW using the LINX toolkit over a USB serial connection. LabVIEW calculates the differences between opposing sensor readings and applies a threshold-based decision rule to select an increase, decrease, or no-change correction for each axis. Numerical indicators and shift registers are used to display measurements and retain the calculated angle-related values. The implemented work demonstrates sensor acquisition, graphical programming, threshold-based decision logic, and microcontroller-computer interfacing. Physical servo positioning, solar-panel integration, and quantitative tracking-efficiency evaluation are identified as further development rather than completed experimental outcomes.

Index Terms—solar tracking, dual-axis control, light-dependent resistor, LabVIEW, Arduino Nano, LINX, virtual instrumentation.

I. INTRODUCTION

Photovoltaic (PV) systems convert incident solar radiation into electrical energy. The orientation of a panel affects how directly sunlight reaches its surface. Because the apparent solar position changes throughout the day and across seasons, a panel held at a fixed angle is not continuously aligned with the incoming light [1], [2].

Single-axis tracking can follow motion along one direction [3], while dual-axis tracking provides adjustment in both horizontal and vertical directions and has been reported to outperform fixed panels [4]. Tracking approaches range from passive mechanisms [5] to active sensor-driven systems. However, commercial tracking systems may require specialised sensors, controllers, and mechanical assemblies. A lower-cost educational prototype can instead use light-dependent resistors

[6] and a graphical programming environment to demonstrate the core sensing and decision-making principles.

This work presents a LabVIEW-based control prototype for a dual-axis smart solar tracker. Four LDRs provide relative illumination measurements, an Arduino Nano [9] acquires the corresponding analog signals, and LabVIEW [7], [8] processes the measurements through the LINX interface [10]. The present implementation focuses on acquisition and calculation of angular corrections; completed physical movement of a solar panel is not claimed.

1. Objectives

The objectives of the project are:

- to detect relative light intensity using four LDRs arranged as left, right, top, and bottom sensors;
- to acquire the sensor signals using the analog inputs of an Arduino Nano;
- to communicate the measurements to LabVIEW through LINX and a USB serial link;
- to calculate horizontal and vertical sensor differences and apply a threshold-based decision rule; and
- to display the acquired measurements and calculated values through a LabVIEW front panel.

1. System Overview

The proposed system consists of four LDR voltage-divider circuits, an Arduino Nano, a USB serial interface, and a Lab-VIEW virtual instrument. The LDRs are placed in opposing pairs around a small cross-shaped divider. When the light source is not centred, the divider casts different shadows on the sensors, producing a difference between opposing readings. The difference indicates the direction in which the sensing assembly should be corrected. The overall system is shown in Fig. 1.

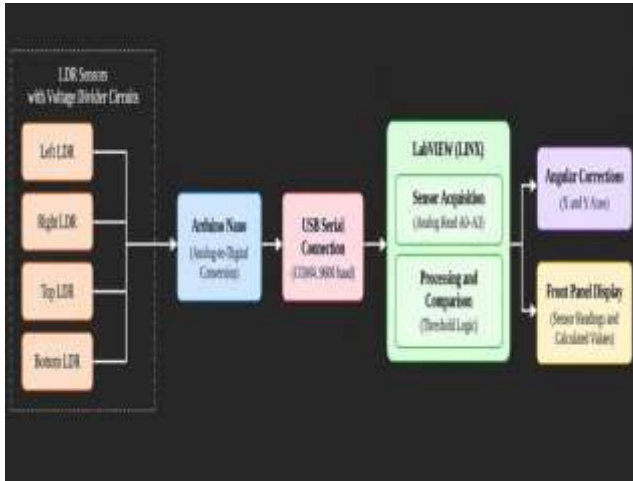


Fig. 1. Overall system block diagram showing the sensing, acquisition, serial link, and LabVIEW processing path.

2. Hardware Components

The Arduino Nano is based on the ATmega328P microcontroller [11], [12]. The principal components and the estimated costs reported for the prototype are listed in Table I. Prices are estimates and can vary by supplier. The assembled sensing and acquisition prototype is shown in Fig. 2.

Table I

Core Hardware Components And Estimated Cost

Component	Qty.	Unit (Rs.)	Total (Rs.)
Arduino Nano	1	400	400
LDR	4	10	40
10 kΩ, 1/4 W resistor	4	2	8

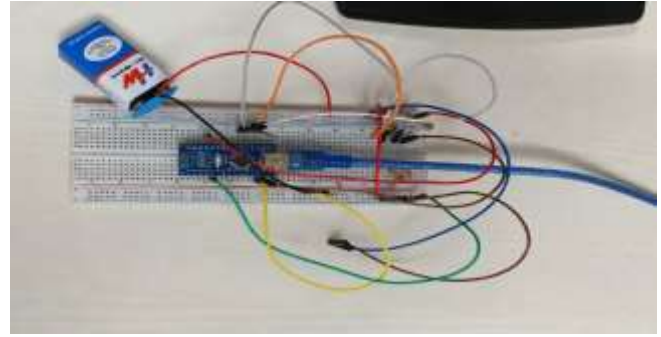


Fig. 2. Assembled sensing and acquisition prototype: Arduino Nano, bread-board with LDR voltage dividers, and USB connection to the computer.

The bill of materials above covers the sensing and acquisition prototype. Servo motors, a positioning bracket, and a mounted solar panel are not included in this total because their integration is proposed as future work in the source project documentation.

3. Sensor Circuit and Operating Principle

An LDR changes resistance as illumination changes; its resistance generally decreases as incident light increases. Since an Arduino analog input measures voltage, each LDR is connected in a voltage-divider circuit with a fixed resistor [6], as shown in Fig. 3. For the configuration in which the LDR is connected to the supply and the fixed resistor is connected to ground, the output voltage is

$$V_{out} = V_S \frac{R_F}{R_L + R_F} \quad (1)$$

where V_S is the supply voltage, R_F is the fixed resistance, and R_L is the LDR resistance. With this arrangement, a decrease in R_L produces an increase in V_{out} .

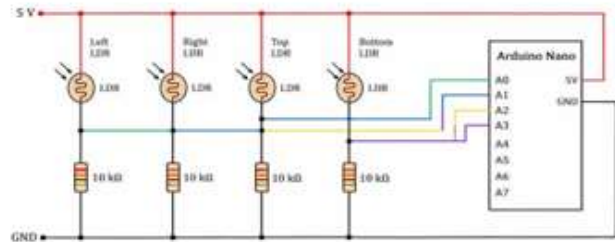


Fig. 3. Circuit diagram of the four LDR voltage dividers connected to the analog inputs of the Arduino Nano.

The four analog readings are labelled VL, VR, VT, and VB, corresponding to the left, right, top, and bottom sensors. The physical wiring must match these labels for the signs of the calculated differences to represent the intended directions correctly.

II. LABVIEW SOFTWARE DESIGN

The complete LabVIEW block diagram of the system is shown in Fig. 4; its stages are described in the following subsections.

1. Microcontroller Communication

The Arduino Nano provides the analog-to-digital acquisition interface. LabVIEW communicates with the board through the LINX toolkit and a USB serial connection, which relies on VISA serial resource handling [13]. The project documentation describes a serial configuration using COM4 at 9600 baud; the actual COM port must be selected according to the host computer's device assignment. LINX initialization is performed before repeated acquisition, and the communication session is closed when the user stops the program.

2. Analog Acquisition

Four LINX Analog Read functions acquire channels 0, 1, 2, and 3, corresponding to Arduino pins A0–A3. The readings are shown on numerical indicators in the LabVIEW front panel. These indicators support debugging by making it possible to observe whether each channel responds when illumination changes.

3. Difference Calculation

The horizontal and vertical differences are calculated as

$$D_X = V_L - V_R \quad (2)$$

$$D_Y = V_T - V_B \quad (3)$$

$$V_{out} = VS$$

where D_X and D_Y represent the horizontal and vertical difference signals, respectively. The subtraction order is important: swapping the inputs reverses the sign and therefore reverses the direction associated with a positive or negative difference.

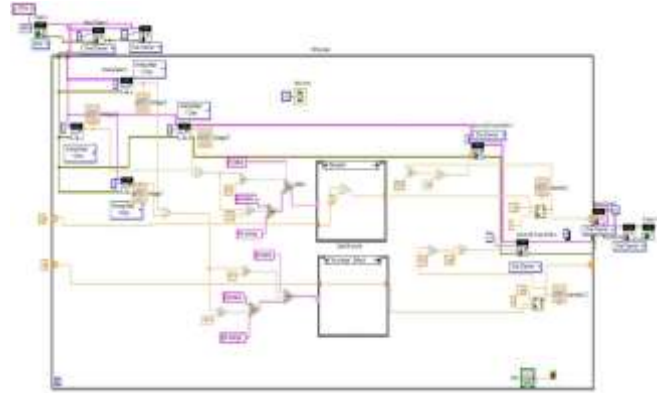


Fig. 4. LabVIEW block diagram of the dual-axis solar tracking system.

4. Threshold-Based Decision Logic

A deadband is used to prevent small measurement differences from continually changing the calculated output. The project uses threshold values of +0.1 and -0.1 for the difference signals. For either axis, the decision rule is

$$u(D) = \begin{cases} \text{Increase,} & D > 0.1, \\ \text{Decrease,} & D < -0.1, \\ \text{No change,} & -0.1 \leq D \leq 0.1. \end{cases} \quad (4)$$

The selected direction must be checked against the actual sensor placement and channel mapping. The threshold is expressed in the units of the acquired readings, so it should be re-evaluated if the signal scaling or acquisition configuration changes.

3. Angle-State Update And Output Processing

Two LabVIEW Case Structures implement the increase, decrease, and no-change cases for the horizontal and vertical calculations. Shift registers retain the angle-related state between iterations and are initialized to 90, as described in the project documentation. The resulting values pass through arithmetic and range-handling functions before being displayed. The documented diagram includes a 100-ms wait in the loop, which limits how rapidly the acquisition-and-decision sequence repeats.

The current project description supports calculation and display of angle-related values. The available documentation does not establish a completed closed-loop test in which servo motors physically reposition a

panel. Consequently, the calculated values are treated as control commands or intermediate outputs, not as measured physical panel angles.

III. ALGORITHM AND EXECUTION FLOW

The program follows the sequence below, which is also summarised by the flowchart in Fig. 5:

1. Initialise the LINX serial communication session.
2. Configure the Arduino interface and initialise angle-related values.
3. Read the four analog channels corresponding to the LDRs.
4. Display the sensor measurements on the
5. Calculate DX and DY using (2) and (3).
6. Compare each difference against the positive and negative thresholds.
7. Select Increase, Decrease, or No change for each axis.
8. Update or retain the angle-related values using the shift registers.
9. Apply the documented output arithmetic and range handling.
10. Repeat until the stop control is activated, then close the LINX session.

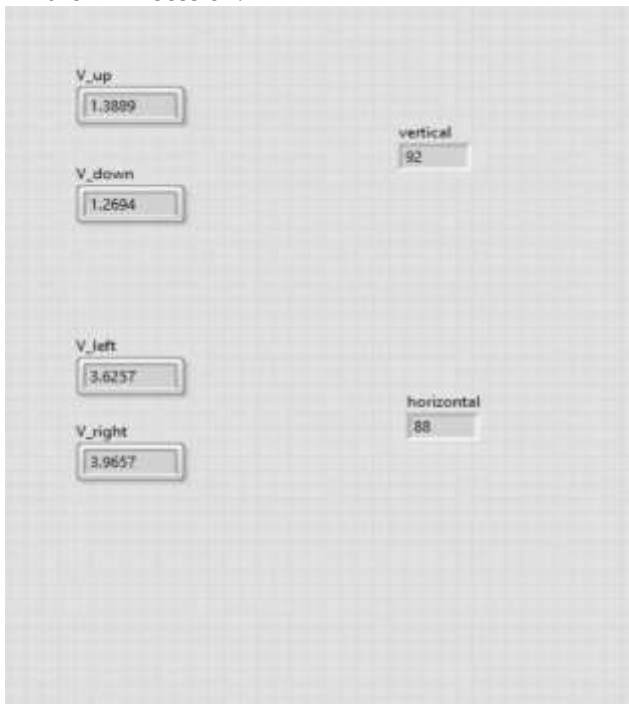


Fig. 5. Flowchart of the LabVIEW program execution.



Fig. 6. LabVIEW front panel displaying the sensor voltages and calculated angle values.

The repeated loop allows the software to respond to changing illumination without restarting the virtual instrument for every measurement.

IV. RESULTS AND DISCUSSION

The project documentation reports implementation of the LabVIEW-based light-sensing and angular-correction calculation path. Four LDR channels are acquired through the Arduino Nano and LINX. The software calculates differences between opposing sensors, evaluates the differences against thresholds of +0.1 and -0.1, and selects Increase, Decrease, or No change for each axis. Shift registers retain angle-related values initialized to 90, while numerical indicators display the acquired and processed values, as seen on the front panel in Fig. 6.

These results demonstrate the intended software workflow: analog acquisition, serial communication, arithmetic processing, threshold comparisons, state retention, and front-panel monitoring. However, the source material does not provide a quantitative dataset for tracking error, response time, power improvement, or alignment accuracy. Such metrics are therefore not reported here. In addition, physical servo positioning and solar-panel integration are described as future extensions, so an increase in photovoltaic energy output has not been experimentally established by the current implementation.

1. Practical Issues

The project documentation identifies several practical issues encountered during testing:

1. COM-port busy: another application or virtual instrument may be using the serial resource. Close the other application and restart the VI.
2. Arduino not detected: verify the USB connection and selected port.
3. Constant or unstable sensor readings: inspect the voltage-divider wiring, analog channel mapping, connections, and ambient-light conditions.
4. No angular-value change: check the sensor-difference values, subtraction order, and threshold conditions.
5. LINX communication error: verify the firmware, port selection, and initialization sequence.

2. Limitations And Future Work

The present implementation is a low-cost sensing and software-processing prototype. Its main limitation is that the supplied project results do not establish physical dual-axis movement or measured improvement in solar energy capture. Several extensions are therefore recommended. First, two servo motors and a suitable pan-tilt bracket can be added to convert the calculated corrections into physical movement. Before connection, the output mapping, angle convention, mechanical limits, and motor calibration should be verified. Second, a small solar panel can be mounted on the mechanism to evaluate the system under real illumination. Third, sensor calibration and averaging or filtering can be investigated to reduce fluctuations caused by noise and changing ambient illumination. An adaptive deadband may also improve stability. Finally, performance can be evaluated by comparing calculated corrections with known light-source directions and, after panel integration, comparing electrical output under equivalent lighting conditions.

V. CONCLUSION

This paper presented the design and implementation of a low-cost dual-axis smart solar tracking control prototype using four light-dependent resistors (LDRs), an Arduino Nano, and NI LabVIEW with the LINX toolkit. The primary objective was to develop a system capable of detecting variations in light intensity along the

horizontal and vertical directions and processing the acquired sensor readings through a graphical programming environment. The prototype demonstrates the integration of sensor interfacing, analog data acquisition, serial communication, and threshold-based decision-making within a virtual instrumentation framework.

The four LDR sensors are connected through voltage-divider circuits to the analog input channels of the Arduino Nano. The acquired readings are transmitted to LabVIEW through a USB serial connection using LINX. Within LabVIEW, the differences between opposing sensor readings are calculated to determine the direction of the required correction along each axis. A threshold-based decision algorithm classifies these differences into increase, decrease, or no-change conditions. Case Structures and shift registers are used to process the decisions and retain the calculated angle-related values between successive iterations. The front panel provides numerical indicators for observing the sensor readings and processed outputs, thereby supporting system monitoring and troubleshooting.

The use of readily available components makes the proposed approach suitable for educational applications involving instrumentation, microcontroller interfacing, and graphical programming. The modular structure of the software also allows individual stages, including sensor acquisition, difference calculation, threshold comparison, and output processing, to be examined and tested independently. The reported hardware component cost of approximately Rs. 648 further demonstrates the affordability of the sensing and acquisition subsystem, although additional components would be required for a complete motorized solar tracker.

Despite these advantages, the present implementation has certain limitations. The current results establish the sensor acquisition and software-processing workflow but do not provide

quantitative measurements of tracking accuracy, response time, or improvement in photovoltaic energy generation. Physical servo positioning and integration of a solar panel must be completed and tested before the performance of the system can be evaluated as a fully operational dual-axis tracker. Sensor calibration,

noise reduction, and refinement of the threshold values would also help improve the reliability of the tracking decisions under changing illumination conditions. Future development will focus on integrating two servo motors with a suitable mechanical mounting arrangement to enable physical movement along both axes. The system can subsequently be evaluated under different lighting conditions by measuring alignment error, response time, and electrical output. A comparison between fixed-panel and tracked-panel operation under equivalent illumination conditions would provide a more objective assessment of the potential benefits of the proposed approach. Additional improvements, such as averaging sensor readings and implementing an adaptive threshold, may further reduce unnecessary corrections caused by small fluctuations in light intensity.

In conclusion, the project establishes a foundation for a cost-effective, LabVIEW-based solar tracking system and demonstrates how simple optical sensors, a microcontroller, and graphical programming can be combined to implement a two-axis light-detection and decision-making process. With further actuator integration, calibration, and experimental validation, the prototype can be extended into a complete solar tracking system and serve as a practical platform for studying automated control and renewable-energy instrumentation.

References

1. J. A. Duffie and W. A. Beckman, *Solar Engineering of Thermal Processes*, 4th ed. Hoboken, NJ, USA: Wiley, 2013.
2. H. Mousazadeh, A. Keyhani, A. Javadi, H. Mobli, K. Abrinia, and A. Sharifi, "A review of principle and sun-tracking methods for maximizing solar systems output," *Renew. Sustain. Energy Rev.*, vol. 13, no. 8, pp. 1800–1818, 2009.
3. S. A. Kalogirou, "Design and construction of a one-axis sun-tracking system," *Solar Energy*, vol. 57, no. 6, pp. 465–469, 1996.
4. R. Eke and A. Senturk, "Performance comparison of a double-axis sun tracking versus fixed PV system," *Solar Energy*, vol. 86, no. 9, pp. 2665–2672, 2012.
5. M. J. Clifford and D. Eastwood, "Design of a novel passive solar tracker," *Solar Energy*, vol. 77, no. 3, pp. 269–280, 2004.
6. J. Rizk and Y. Chaiko, "Solar tracking system: more efficient use of solar panels," *World Acad. Sci., Eng. Technol.*, vol. 41, pp. 313–315, 2008.
7. J. Travis and J. Kring, *LabVIEW for Everyone*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2006.
8. National Instruments, "LabVIEW Documentation." [Online]. Available: <https://www.ni.com/en/support/documentation.html>
9. Arduino, "Arduino Nano – Official Documentation." [Online]. Available: <https://docs.arduino.cc/hardware/nano/>
10. MakerHub, "LINX Toolkit for LabVIEW." [Online]. Available: <https://www.ni.com/en/support/downloads/tools-network/download.labview-hobbyist-toolkit.html>
11. Microchip Technology Inc., "ATmega328P 8-bit AVR Microcontroller Datasheet." [Online]. Available: <https://www.microchip.com/en-us/product/ATmega328P>
12. M. Banzi and M. Shiloh, *Getting Started with Arduino*, 3rd ed. Sebastopol, CA, USA: Maker Media, 2014.
13. National Instruments, "NI-VISA Documentation." [Online]. Available: <https://www.ni.com/en/support/documentation.html>