Arduino Solar Tracker

May 20, 2023

Ahmed El Nokrashy
900181162
Spring 2023
PHYS 4241 Introduction to Solar Energy

1. Introduction

The purpose of this project report is to present the design, development, and implementation of an Arduino-based solar tracker. The solar tracker aims to maximize solar panels' efficiency by aligning them with the sun's direction throughout the day. The system utilizes an Arduino microcontroller, light sensors, and servo motors to automatically adjust the position of the solar panels.

2. Project Overview

2.1. Components

The solar tracker consists of the following key components:

1. Arduino Uno: The Arduino Uno serves as the main control unit of the system, responsible for processing sensor inputs and controlling the movement of the servo motors.

Figure 1: Arduino UNO Microcontroller

2. Light Sensors: Light-dependent resistors (LDRs) are used as light sensors to detect the intensity of sunlight. Four LDRs are mounted in a cubicle formation above the solar panel structure to capture light levels from different angles.

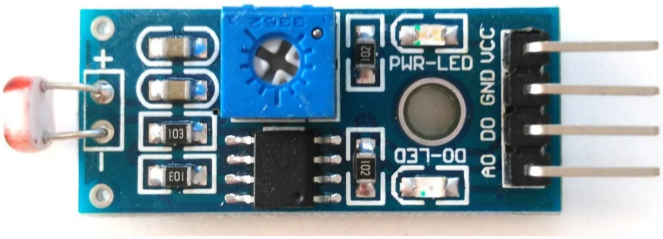

Figure 2: Light Dependent Resistor

3. Servo Motors: Two servo motors are connected to the solar panel structure to control its azimuth and elevation angles.
   - MG996R Servo: Used to control the base mount for horizontal XY axis motion from an initial position of 90 degrees.It can move 90 degrees clockwise or anticlockwise.

Figure 3: MG996R Servo

   - MG90S Micro servo: Used to control the panel vertical orientation with an initial postion at 90 degrees parallel to Z axis. It can move 90 degrees clockwise or anticlockwise.

Figure 4: MG90S Micro Servo

4. Solar Panels: The solar panels are mounted on a structure that can be tilted and rotated by the servo motors.

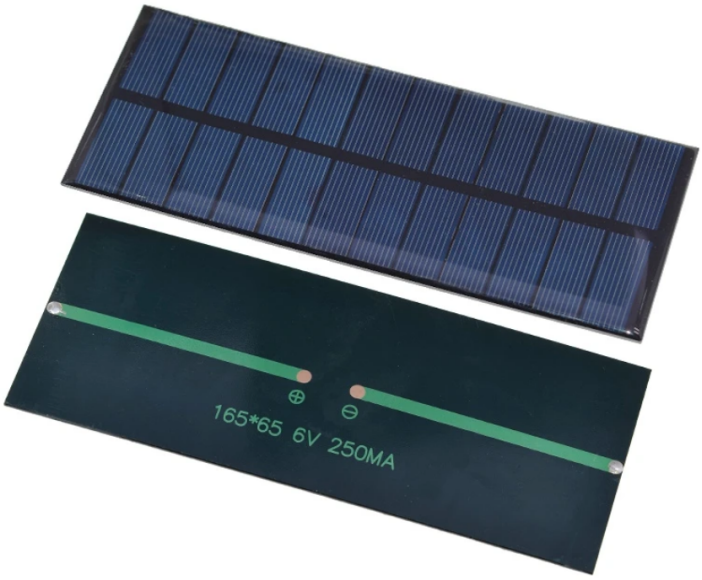

Figure 5: Mini Solar Panel 6V/250mA

6. A breadboard is used for wire organization and easier debugging.

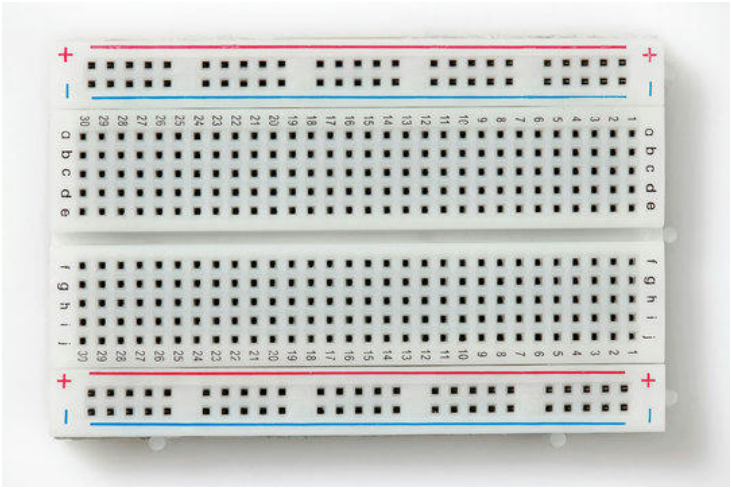

Figure 6: Mini Breadboard for wire management

5. Power Supply: A suitable power supply is provided to ensure the continuous operation of the Arduino and servo motors.

- Could be powered directly through the USB port. (not portable)
- Could be powered through a 9-volt battery installed via a clip.

2.2. Design and Implementation:

2.2.1. Control

The control sequence was first implemented and tested on TikerCad circuit tool.

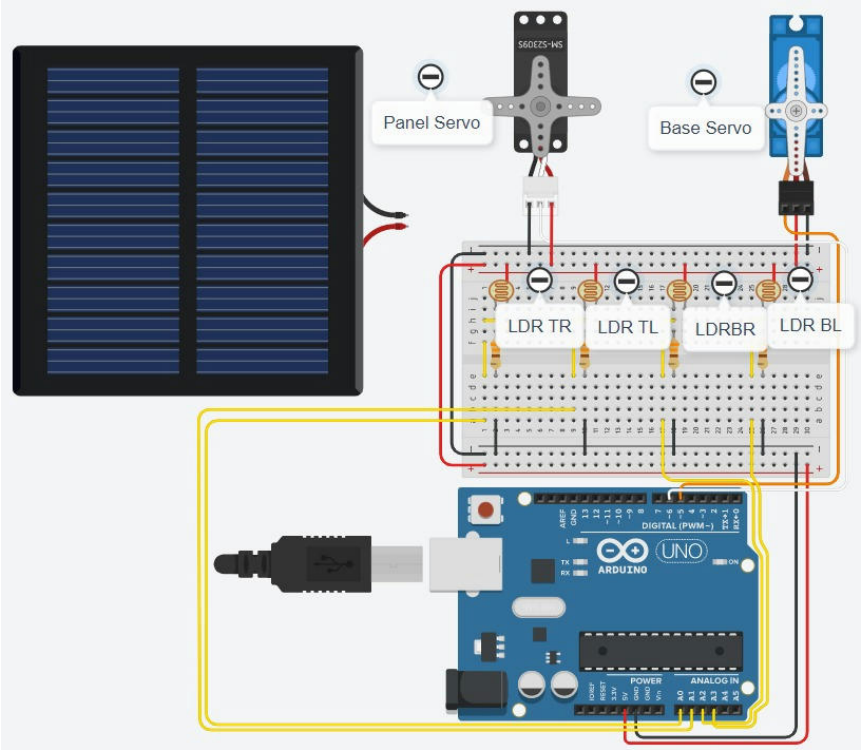

Figure 7: Control Schematic

2.2.2. Hardware Design

The structural was modeled on SolidWorks to ensure accurate dimensions during assembly, and genrealy make my life easier.

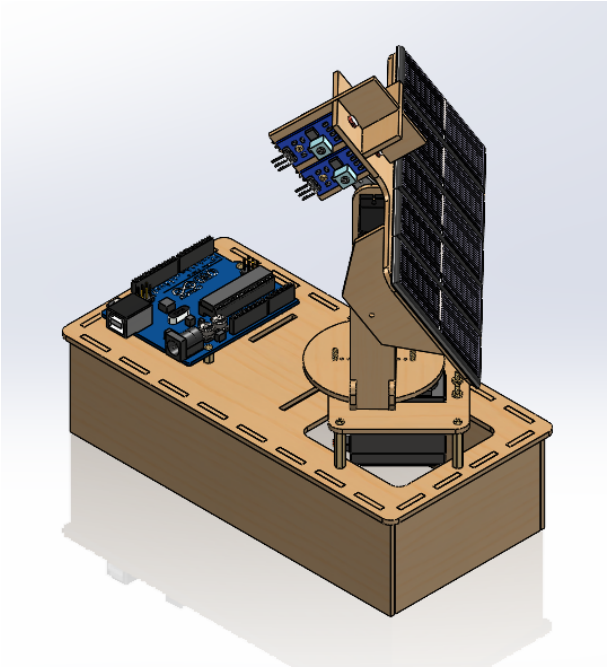

Figure 8: Solar Tracker Solid works Model

1. The solar panel structure is designed to hold the solar panels securely and allow movement in both azimuth and elevation directions.
2. Two servo motors are attached to the structure, with one controlling the azimuth movement and the other controlling the elevation movement.
3. Light sensors (LDRs) are positioned on the structure to detect the intensity of sunlight from different angles.
4. The Arduino Uno is connected to the LDRs and servo motors through appropriate wiring and connectors.

2.2.3. Software Design:

Code Documentation:

- It starts by including the Servo library, which provides functions to control the servo motors.
- It defines and initializes variables for the horizontal and vertical servos and their initial positions and limits.
- It defines variables for connecting the LDRs to specific pins on the Arduino board.
- The setup() function is called once at the beginning of the program. It initializes the serial communication at a baud rate of 9600 and attaches the horizontal and vertical servos to the corresponding pins. Then, it sets the initial position of the servos to 90 degrees and introduces a delay of 3 seconds (3000 milliseconds).
- The loop() function is called repeatedly after the setup() function. It performs the main logic of the program.
- It defines boolean variables update_horizontal and update_vertical to keep track of whether the horizontal and vertical servos need to be updated.
- It reads the analog values from the LDRs and stores them in variables tr, tl, br, and bl (top right, top left, bottom right, bottom left).
- It calculates the average values for the top, bottom, left, and right LDRs.
- It calculates the differences between the average values of the top-bottom and left-right LDR pairs.
- It prints the value of dhoriz (horizontal difference) to the serial monitor.
- It checks if the difference in the vertical LDR values exceeds the tolerance (tol). If the difference is greater than the tolerance, it adjusts the servov variable (vertical servo position) accordingly. If avt (average top) is greater than avd (average bottom), it increments servov unless it reaches the upper limit (servovLimitHigh). Similarly, if avt is less than avd, it decrements servov unless it reaches the lower limit (servovLimitLow). The update_vertical flag is set to true if the vertical servo position needs to be updated.
- It checks if the difference in the horizontal LDR values exceeds the tolerance. If the difference is greater than the tolerance, it adjusts the servoh variable (horizontal servo position) accordingly. If avl (average left) is greater than avr (average right), it increments servoh unless it reaches the lower limit (servohLimitLow). Similarly, if avl is less than avr, it decrements servoh unless it reaches the upper limit (servohLimitHigh). The update_horizontal flag is set to true if the horizontal servo position needs to be updated.

- If update_vertical is true, it sets the position of the vertical servo using the write() function and the servov variable.
- If update_horizontal is true, it sets the position of the horizontal servo using the write() function and the servoh variable.
- It introduces a delay of dtime milliseconds before starting the loop again.

3. Testing and Results

1. The solar tracker system was tested and working properly for its scope and servo design.
2. The system demonstrated acceptable tracking and alignment with the sun, maximizing the energy output of the solar panels.
3. Performance metric, such as voltage output, is observed through a voltmeter.

4. Conclusion

The Arduino-based solar tracker successfully achieved its objective of maximizing the efficiency of solar panels by tracking the sun's position. The system's design, implementation, and testing effectively ensured reliable operation for the scope it was designed at.