Project on

Automatic Solar Tracker

1. **Introduction**

Solar energy is becoming more and more prevalent across the world. Currently, many methods are being researched to make solar panels output more energy, reducing our reliance on fossil fuels and coal. One way to do this is to have the panels move, always facing the sun in the sky. This allows optimal energy collection, making solar panels more efficient.

This Instruct able will look into how solar trackers work, and implement such a method into a solar tracker prototype using an Arduino UNO.

**1.1 Working**

There are 3 main methods which are used to control a solar tracker. The first is a passive control system, and the other two are active control systems. The passively controlled solar tracker contains no sensors or actuators but changes its position based on heat from the Sun. By using gas with a low boiling point in a container mounted on hinges at its middle, similar to a see-saw, the solar panel can change its position based on the direction of heat from the Sun.

The active systems are a bit different. Both require a processing system, as well as actuators to move the panels. One way to actively control solar panels is to transmit the Sun's position to the panels. The panels then orient themselves to this position in the sky. Another method is by using sensors to detect the sun's position. By using Light Dependent Resistors (LDRs), its possible to detect varying light levels. These sensors are then used to determine where the sun is in the sky, allowing the panel to orient itself appropriately.

In this Instructable, we will use the sensor based active control system.

**1.2 Purpose**

Solar tracker, a system that positions an object at an angle relative to the Sun. The most-common applications for solar trackers are positioning photovoltaic (PV) panels (solar panels) so that they remain perpendicular to the Sun’s rays and positioning space telescopes so that they can determine the Sun’s direction. PV solar trackers adjust the direction that a solar panel is facing according to the position of the Sun in the sky. By keeping the panel perpendicular to the Sun, more sunlight strikes the solar panel, less light is reflected, and more energy is absorbed. That energy can be converted into power.

**1.3 Scope**

Solar trackers provide significant advantages for renewable energy. With solar tracking, power output can be increased by about 30 to 40 percent. The increase in power output promises to open new markets for solar power. However, solar trackers have several important disadvantages. A static solar panel may have a warranty that spans decades and may require little to no maintenance. Solar trackers, on the other hand, have much shorter warranties and require one or more actuators to move the panel. These moving parts increase installation costs and reduce reliability; active tracking systems may also use a small amount of energy (passive systems do not require additional energy). Computer-based algorithm solar trackers are more expensive, require additional maintenance, and become obsolete much faster than static solar panels, since they use fast-evolving electronic components with parts that may be difficult to replace in relatively short periods of time.

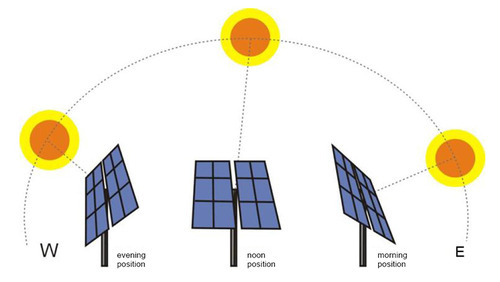


Figure 1.1 Solar Tracker

1. **Literature Survey**

**Arduino Programming by Mark Torvalds.**We have referred this to understand about the fundamentals of Arduino programming. We understood many functions and syntax that are essential for our Arduino IDE code.

**Fundamentals of Electronics by Behzad Razavi.**We have referred this to understand the basic components of electronics like leds, resistor ,BJTs ,MOS . Also we have seen the designing steps which are given in detail in the form of examples.

**Exploring Arduino by Jeremy Blum**. We have referred this to understand about the fundamentals and experiments of Arduino Programming. We understood many Factors that can be helpful for programming the code.

**Light Intensity Monitoring system by Gajjala Ashok.** We have referred to this research paper to understand about the basics of our light monitoring. We have also understood the I-O process of light monitoring.

**Light Dependent Resistance by Derci Felix da Silva.** We have referred to this research paper to understand the fundamentals of light dependent resistor. We have also understood the factors responsible for change in resistance of LDR due to different light intensities.

1. **Project Plan**

**3.1 Cost Estimation**

The overall project cost has been estimated taking into account the various other costs that had been encountered en-routed to development of the project.

These costs include:

* Especially, the cost of electronic component namely Arduino UNO, LDR (5 MΩ), resistors (10k Ω Miscellaneous Costs: This cost includes the expenses incurred for various miscellaneous reasons other than those specified above.
* Total Estimated Cost Rs.1500.

**3.2 Action Plan**

* First we place and align our arudino uno
* Then we will connect it with solar panel which is directly connected with motor.
* We program arudino uno in such a way that it detects the intensity of light.

* The intensity of light acts as a source of rotating of solar tracker in the higher intensity region.

* Motor connected acts as a source to turn the solar panel with respect to light

* After that inject the source code in Arduino Uno.

* All done. Automatic solar tracker is ready to use.

1. **Architecture**

**4.1 Hardware Implementation**

1. **Arduino Uno**



Fig 4.1 Arduino Uno

Arduino is an open-source electronics platform based on easy-to-use hardware and software. [Arduino boards](https://www.arduino.cc/en/Main/Products) are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the [Arduino programming language](https://www.arduino.cc/en/Reference/HomePage) (based on [Wiring](http://wiring.org.co/)), and [the Arduino Software (IDE)](https://www.arduino.cc/en/Main/Software), based on [Processing](https://processing.org/).

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of [accessible knowledge](http://forum.arduino.cc/) that can be of great help to novices and experts alike.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The [software](https://www.arduino.cc/en/Main/Software), too, is open-source, and it is growing through the contributions of users worldwide.

1. **LDR (Light Dependent Resistor)**



Fig 4.2 Light Dependent Resistor

Light dependent resistors, LDRs or photoresistors are often used in electronic circuit designs where it is necessary to detect the presence or the level of light.

These electronic components can be described by a variety of names from light dependent resistor, LDR, photoresistor, or even photo cell, photocell or photoconductor.

Although other electronic components such as photodiodes or photo-transistor can also be used, LDRs or photo-resistors are a particularly convenient to use in many electronic circuit designs. They provide large change in resistance for changes in light level.

In view of their low cost, ease of manufacture, and their ease of use, LDRs have been used in a variety of different applications. At one time LDRs were used in photographic light meters, and even now they are still used in a variety of applications where it is necessary to detect light levels.

1. **LEDs**

Fig 4.3 LED (Light Emitting Diode)

In the simplest terms, a light-emitting diode (LED) is a semiconductor device that emits light when an electric current is passed through it. Light is produced when the particles that carry the current (known as electrons and holes) combine together within the semiconductor material.

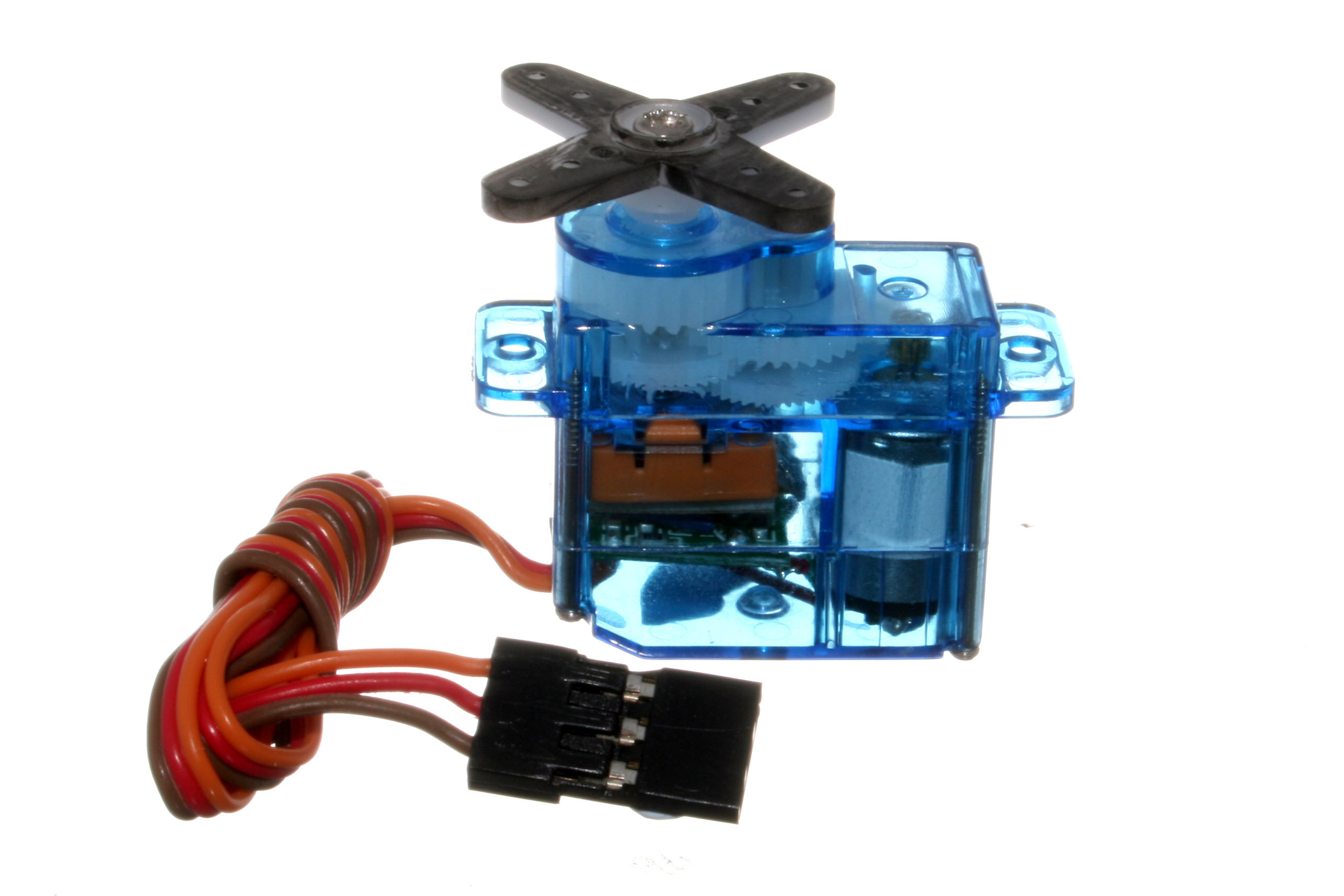
Since light is generated within the solid semiconductor material, LEDs are described as solid-state devices. The term solid-state lighting, which also encompasses organic LEDs (OLEDs), distinguishes this lighting technology from other sources that use heated filaments (incandescent and tungsten halogen lamps) or gas discharge (fluorescent lamps).

The main semiconductor materials used to manufacture LEDs are:

* Indium gallium nitride (InGaN): blue, green and ultraviolet high-brightness LEDs
* Aluminum gallium indium phosphide (AlGaInP): yellow, orange and red high-brightness LEDs
* Aluminum gallium arsenide (AlGaAs): red and infrared LEDs
* Gallium phosphide (GaP): yellow and green LEDs

1. **Servos**

Servo motors have been around for a long time and are utilized in many applications. They are small in size but pack a big punch and are very energy-efficient. These features allow them to be used to operate remote-controlled or radio-controlled toy cars, robots and airplanes. Servo motors are also used in industrial applications, robotics, in-line manufacturing, pharmaceutics and food services. But how do the little guys work?



The servo circuitry is built right inside the motor unit and has a positionable shaft, which usually is fitted with a gear (as shown below). The motor is controlled with an electric signal which determines the amount of movement of the shaft.

1. **Resistor**

**A picture containing indoor

Description automatically generated**

Fig 4.6 Resistors

A resistor is a passive two terminal electrical component that implements electrical resistanceas a circuit element. In electronic circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, biasactive elements, and terminate transmission lines, among other uses.

Resistors are common elements of electrical networks and electronic circuitsand are ubiquitous in electronic equipments. Practical resistors as discrete components can be composed of various compounds and forms. Resistors are also implemented within integrated circuits

The electrical function of a resistor is specified by its resistance: common commercial resistors are manufactured over a range of more than nine orders of magnitude.

**4.2 Software Implementation**

* 1. **Arduino IDE**

The Arduino [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) is a [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) application (for [Windows](https://en.wikipedia.org/wiki/Windows), [macOS](https://en.wikipedia.org/wiki/MacOS), [Linux](https://en.wikipedia.org/wiki/Linux)) that is written in the programming language [Java](https://en.wikipedia.org/wiki/Java_(programming_language)). It originated from the IDE for the languages [Processing](https://en.wikipedia.org/wiki/Processing_(programming_language)) and [Wiring](https://en.wikipedia.org/wiki/Wiring_(development_platform)). It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, [brace matching](https://en.wikipedia.org/wiki/Brace_matching), and [syntax highlighting](https://en.wikipedia.org/wiki/Syntax_highlighting), and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus. The source code for the IDE is released under the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License).

The Arduino IDE supports the languages [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B) using special rules of code structuring. The Arduino IDE supplies a [software library](https://en.wikipedia.org/wiki/Software_library) from the [Wiring](https://en.wikipedia.org/wiki/Wiring_(development_platform)) project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable [cyclic executive](https://en.wikipedia.org/wiki/Cyclic_executive) program with the [GNU toolchain](https://en.wikipedia.org/wiki/GNU_toolchain), also included with the IDE distribution. The Arduino IDE employs the program to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

The source code for the IDE is released under the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License), version 2.The Arduino IDE supports the languages [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B) using special rules of code structuring. The Arduino IDE supplies a [software library](https://en.wikipedia.org/wiki/Software_library) from the wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable [cyclic executive](https://en.wikipedia.org/wiki/Cyclic_executive) program with the [GNU toolchain](https://en.wikipedia.org/wiki/GNU_toolchain), also included with the IDE distribution. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

1. **Project Layout**

****

Fig 5.1.1 Project Layout

**5.1 Project Simulation**

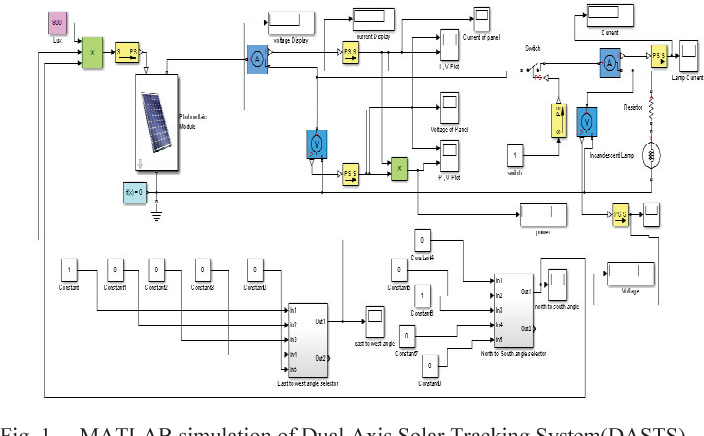
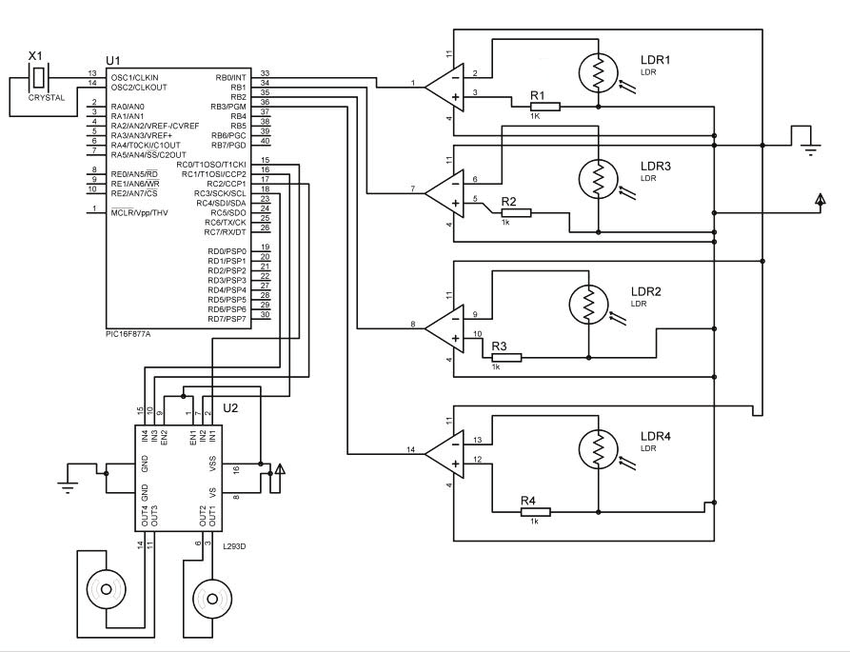


Fig 5.1.2 Simulation

**5.2 Circuit Diagram**

Fig 5.2 Circuit Diagram

**5.3 Final Project Image**

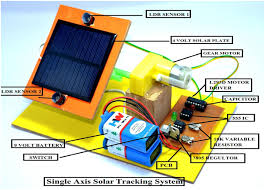


Fig 5.3 Final Project Image

1. **Conclusion**

The study findings showed that the use of low power servo motors carefully designed to move the tracking device hub substantially reduced the power required by the tracking system. Arduino board which belongs to a family of low cost control system was also found to be extremely energy efficient. Multiple-axis solar tracking algorithm improved the power generated by the solar panel with an efficiency improvement of 23.95 %. The study concluded that it is highly important to track the sun at a high degree of accuracy to improve solar irradiation absorption. The developed algorithm has the advantage of space savings compared to adding more solar panels to improve energy generated. The use of Arduino board reduced the system construction cost without depending on complicated, costly and space consuming proprietary infrastructures.

1. **Bibliography**

* **Research Papers**

1. Arduino Programming by Mark Torvalds,
2. Exploring Arduino by Jeremy Blum.
3. Light Intensity Monitoring system by Gajjala Ashok

* **Web**
* [www.google.com](http://www.google.com)
* <https://www.arduino.cc/>
* [www.ieeexplore.ieee.org](http://www.ieeexplore.ieee.org)
* [www.ijser.org](http://www.ijser.org)