**College of Engineering**

**Mechanical and Mechatronics Engineering Department**

**MECA 440- Microcontrollers for Mechatronics**

**Fall 2023**

**Final Project**

PID Light Tracker using Arduino

Rami Kronbi - 20210359

Wassim Ghaddar - 20210586

Dr. Ahmad Koubeisi

[December \*\*, 2023]

Table of Contents

[Project Description 3](#_Toc153235315)

[Functional Specifications 3](#_Toc153235316)

[Block Diagram 5](#_Toc153235322)

[Principle of Operation 6](#_Toc153235323)

[Schematic Diagram 8](#_Toc153235330)

[Bill of Quantity 8](#_Toc153235331)

[Results and Conclusion 9](#_Toc153235332)

[Appendix 9](#_Toc153235333)

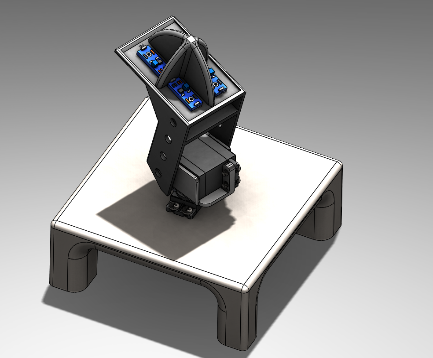
# Project Description

The goal of this project is to build a robot capable of autonomously tracking a light source using PID (Proportional-Integral-Derivative) control. The robot will be equipped with two servo motors that control its yaw and pitch movements, allowing it to adjust its orientation and track the light source accurately. The system will include components such as a robot chassis, servo motors, a light sensor, a microcontroller, and a power supply. The microcontroller will process the sensor readings, calculate the appropriate servo movements using the PID control algorithm, and generate the control signals for the servo motors. By implementing this project, you will create a robot that can track a light source with precision and learn about concepts such as PID control, sensor integration, and servo motor control.

This light-tracking robot project offers a practical application of PID control principles and robotics. The robot's ability to track a light source using servo motors and a light sensor showcases the integration of hardware and software components. Through the development process, you will gain insights into assembling the robot chassis, attaching servo motors, connecting the light sensor, and programming the microcontroller. Additionally, you will have the opportunity to calibrate and fine-tune the PID parameters to optimize the tracking performance. This project provides a hands-on learning experience in control systems, sensor integration, and robotics, enabling you to deepen your understanding of these concepts and explore further applications in the field of robotics and automation.

# Functional Specifications

1. **Chassis Design:**
   * The robot chassis was designed to accommodate both servo motors, light sensor modules, and the microcontroller, along with the wire managing techniques.
   * The material chosen for the design was 3d printed PLA( Polylactic acid) as it offers a lightweight yet sturdy build to allow for easy movement of the Servo Motors without putting too much load on the gears of the motors along with stable directing of the tilt tracking mechanism.



* + The light sensor modules had to be physically on the surface of the mechanism to fine tune their sensitivity via the potentiometer knob on the module as well as not hindering the proper receiving of the signal from the light source.

Figure 1: 3d Model of The Design

# Servo Motors:

* + Two servo motors will be used for controlling the yaw and pitch movements of the robot.
  + The servo motors had to ensure sufficient range of motion to cover a wide field of view for effective light source tracking. The motors must be responsive and precise in their movements.
  + That’s why the choosing of 2 Mg995 Servo Motor which deliver higher torque values along with precise angle movements capable of the range of motion.

# LDR Sensor Modules

* + The light sensor will be responsible for detecting the intensity of the light source.
  + The capability of providing accurate readings that correspond to the light intensity detected by the sensor was crucial. Thus, the choosing of LDR modules was a better option than normal LDR sensors as it offers the variability in the sensor’s sensitivity and allows for optimum calibration under any conditions.

# Microcontroller

* + The choice of microcontroller was dependent on easiness of control, programming, and applying fundamental PID control algorithms, therefore the choice of the Arduino Uno was a better choice than for example and Arduino nano which sometimes causes unexpected problems.

# PID Control Algorithm

* + The PID control algorithm was implemented to calculate the optimum angle movements for the servo motors based on the difference of the averages of the top and bottom sensors along with the right and left sensors. This ensured proper functionality and produced the most accurate readings from the sensors. Especially, that this system works on 2 Degrees of Freedom and any slight deviation or inaccuracies might cause the system to respond differently than intended to.
  + Fine-tuning parameters of PID( proportional , integral, derivative) gains was done to achieve smoothness and fluidity of the system’s functionality.

# Power Supply

* + An external power supply was used to provide stable and sufficient voltage for the Servo Motors, LDR modules, and the Arduino UNO microcontroller.
  + The choice of an external power supply was important to ensure stable voltage was being delivered to the LDR’s and to eliminate any chance of disturbances due to technical restrictions.

# Block Diagram

**LDR module**

**LDR module**

**LDR module**

**LDR module**

**Arduino UNO**

**Mg995 Servo Motor**

**Mg995 Servo Motor**

**Power Supply**

# Principle of Operation

The core principle lies in harnessing the power of PID (Proportional-Integral-Derivative) control with a dual-axis system. Equipped with servo motors governing yaw and pitch movements, this autonomous system is designed to dynamically adjust its orientation, ensuring seamless tracking of a light source.

This project introduces many techniques in controlling, adjusting, and integrating the system to reduce the percentage of error as much as possible. The principle of operation begins with:

# Light Sensing

* The light sensor plays the pivotal role as the sensory apparatus of the system. Functioning as an optical transducer, it converts incident light intensity into an electric signal.
* This signal is a numeric reflection of the amount of light intensity of the surrounding
* This signal is an analog value which is allows for easier differentiation in the amount of light intensity being provided across each of the four LDR sensors as they are provided in bits (0 to 1023) which governed the resolution of the Analog-to-Digital converter in the Arduino Microcontroller.

# Crossing Operation

Before the Microcontroller engages in the task of error calculation, it first executes an operation known as crossing.

* Crossing is an operation that involves retrieving data from the four sensors that are positioned precisely on the top plate and calculating the average light intensity that crosses spatially between these exact distributed points
* Keeping in mind this system achieves a level of stability in ambient light conditions just as any other luminous conditions and this is due to the crossing technique which calculates difference in average between all four of the sensors
* Averaging helps the system to be adaptable to any light condition and equips the system with the ability to understand the ambient light distribution,
* By providing a more representative value of the overall luminosity surrounding of the robot, accurate, precise, and promising results could be achieved.

# Error Calculation

The essence of precision in the light-tracking robot project lies in the microcontroller's adept handling of error computation. The microcontroller calculates the error based on the results obtained from the crossing technique mentioned above.

* The microcontroller calculates the **ERROR** by subtracting the desired light intensity which is the **SETPOINT** from the actual intensity being calculated from the average being calculated during the crossing method .
* The error represents the deviation of the system from the optimal condition
* The optimal condition in our case is getting as much light intensity being directed towards the sensor since the difference in averaging would be minimal and error would tend to zero
* Corrective actions are taken later PID control Algorithm to achieve this optimal condition

# PID Control Algorithm

* **Proportional (P):** The P component or (**Kp**) scales the error directly to determine the required correction. A higher error results in a larger corrective response. This component is crucial for initial and rapid adjustments, and it is the usually referred to as the ***GAIN***
* **Integral (I):** The I component or (**Ki**) accumulates the error over time, addressing any persistent deviation from the setpoint. It helps eliminate steady-state error caused by the gain being delivered and ensures long-term stability in the system.
  + Note: This component was adjusted in a way to not accumulate any error for more than 10 seconds due to saturation reasons.
* **Derivative (D):** The D component or (**Kd**) anticipates the future trend of the error by considering its rate of change. This helps prevent overshooting and oscillations, enhancing the system's response.

# Servo Motor Control

* After error calculation and applying principal PID control algorithm, the microcontroller translates the output from the PID algorithm into control signals for the servo motors.
* These Signals are known as PWM ( Pulse-Width-Modulation)
* These signals determine the angle or position adjustments required for both yaw and pitch movements based on retrieved values from the PID algorithm.

# Principle of Operation Summary

By maintaining this closed-loop feedback system, the robot adapts to changing conditions, ensuring robust and accurate light tracking. The PID algorithm's dynamic nature, incorporating both immediate and accumulated corrective measures, enhances the overall performance of the system, making it responsive and adaptable in real-world scenarios.

# Schematic Diagram

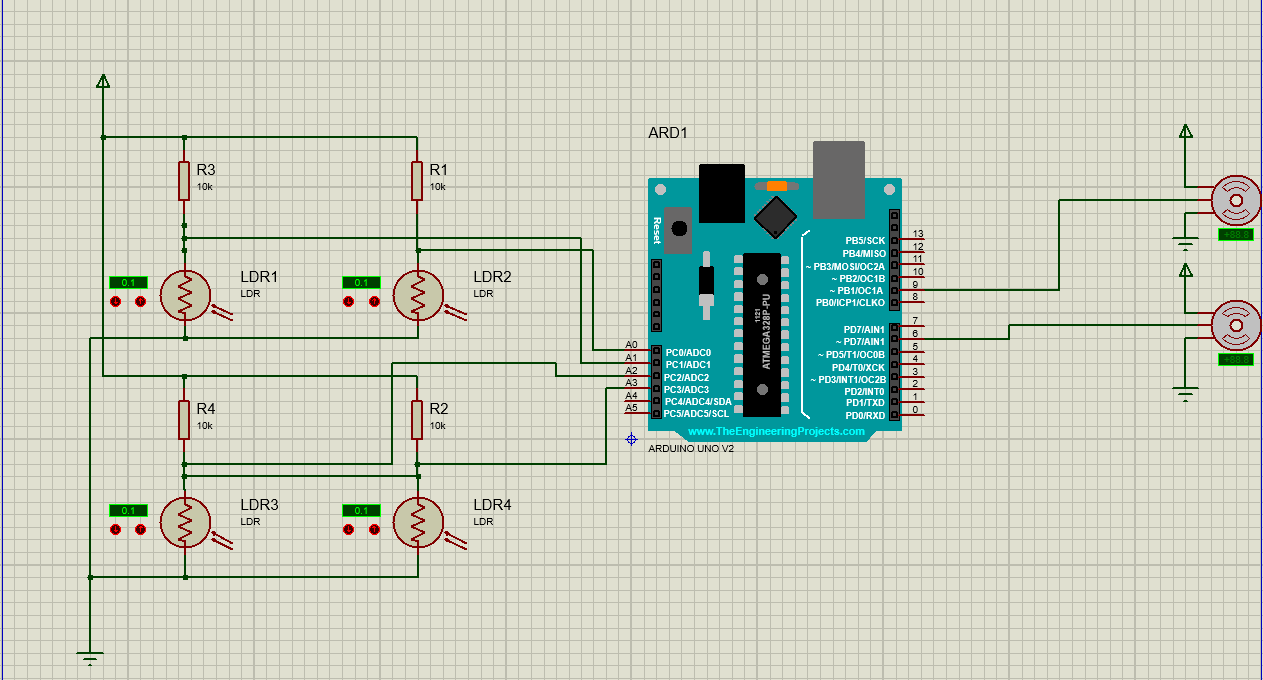


Figure 2: Schematic Diagram showing the Circuitry on Proteus

# Bill of Quantity

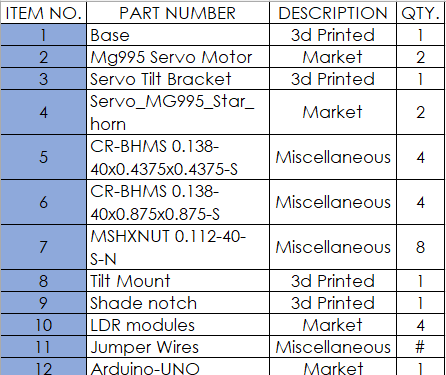
****

Figure 3: 3d Model of The Design

# Results and Conclusion

# Appendix