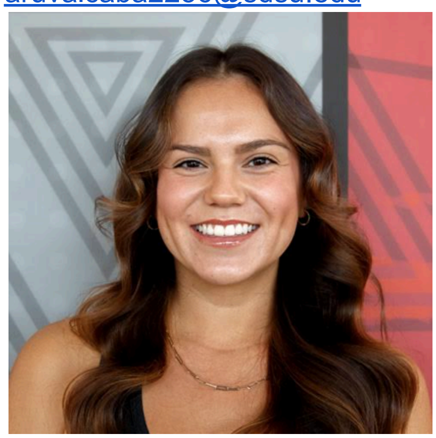
PROJECT TITLE: Automated Light Controller
Student / Team Information:

<table>
<tr><td>Team Name:
Team # on Canvas:</td><td>Angelina/Andrea
2</td></tr>
<tr><td>Team member 1
(Lastname, Firstname; SDSU email; picture):</td><td>Mom, Angelina; amom1053@sdsu.edu
</td></tr>
<tr><td>Team member 2
(Lastname, Firstname; SDSU email; picture):</td><td>Ruvalcaba, Andrea;
aruvalcaba2256@sdsu.edu
</td></tr>
</table>

ABSTRACT (15 points)
(Summarize your project (motivation, goals, system design and results). Max 300 words).

Many individuals find overhead lighting harsh and prefer more adaptable illumination. Current solutions often require manual adjustments, either through physical switches or mobile applications, adding inconvenience. This project addresses this by developing an inexpensive adaptive lighting system that automatically adjusts its brightness based on the ambient light levels in a room, enhancing user comfort and reducing the need for manual intervention. The primary goal is to create a user-friendly, energy-conscious system that can learn and respond to the environment without requiring user attention. The system utilizes an ESP32 microcontroller connected to four photoresistors to measure the light level of the room. The ESP32 averages the readings and controls the brightness of the LED, dimming or brightening

it based on a user-defined threshold (threshold is always automatically set to 1245, but user has the ability to modify as desired). The light level readings, along with the LED brightness and user preferences, is sent to the AWS Cloud Server via Wi-Fi to record the data. The implemented system automatically adjusts the LED's brightness in response to changing light levels of the room, and the AWS Dashboard provides a graph of the room's light level, allowing users to see how the light level changes over time. This automated, cloud-connected system aims to provide a more comfortable and energy-efficient lighting solution that adapts to the user's environment without requiring constant manual adjustments

INTRODUCTION (15 pts)

Motivation/Background (3 pts)

(Describe the problem you want to solve and why it is important. Max 300 words).

Many people dislike overhead lighting. On the market, most solutions to excessive lighting include the need for a manual adjustment such as physically dimming or brightening your light (either physically with a light switch or through an app), or are not cost friendly. We wanted to create a solution where the light automatically adjusts itself based on existing light conditions in the space where the light is. Our solution provides users more ease, less hassle, and extra comfort.

Project Goals (6 pts)

(Describe the project general goals. Max 200 words).

The primary goal of our project is to provide an inexpensive adaptive lighting system that, although can accept user input for preferences, has the ability to work in an adaptive manner that will change the brightness of its light based on the overall room light levels. This system not only benefits the user in a physical aspect but also supports energy consumption efforts and offers a very user-friendly experience that does not need attention or effort. Since it uses cloud connectivity it also stores the user's light preferences and can set itself to the appropriate light level depending on the time.

Assumptions (3 pts)

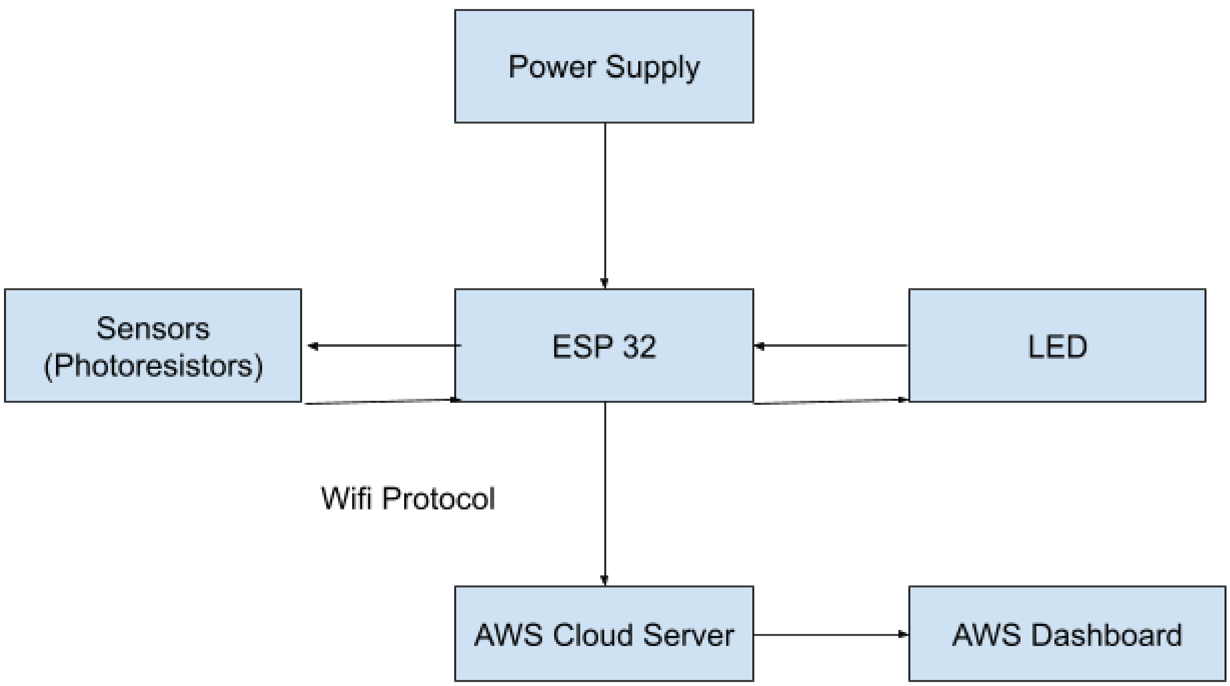
(Describe the assumptions (if any) you are making to solve the problem. Max 180 words).

We can assume the user wants a consistent light level for their space. With that, they want their adaptable system to be wallet friendly, environmentally friendly, but most of all, hassle free and easy to use and implement into any space.

SYSTEM ARCHITECTURE (20 pts)

(Describe the final architecture you have implemented listing sensors, communication protocols (Wi-Fi, BLE, ...), cloud services and user interfaces. Include a block diagram of the system. Max 300 words).

The system architecture implemented in this project centers around the ESP32 microcontroller, which is powered by a PC. Four photoresistors are connected to the ESP32 to capture individual light level readings within the room. This data is then processed by the ESP32, which calculates the average light level internally in the code. This average is used to control an LED's brightness; the LED will dim or brighten based on whether the average light level crosses a user-defined threshold. It dims and brightens in increments of 10 The average light level of the room, the current brightness of the LED, and the user's preferred light level are sent to the AWS Cloud Server using the Wi-Fi protocol to record the data. The AWS Cloud Server works as a bridge for the AWS Dashboard. This AWS Dashboard serves as the user interface and provides a data visualization of the recorded light levels and the user's preferred light level. The user in real-time is able to visually see and monitor the values, providing a better understanding of light fluctuations throughout the day. This software architecture allows for a smooth integrations of the data being collected and processed through our ESP32 and cloud-based visualization using Amazon Web Services. In this SWA, our ESP32 integration acts as the firmware layer. The communication layer is presented by the Wi-Fi and HTTP site. Through the AWS dashboard, we are seeing the presentation layer.



FINAL LIST OF HARDWARE COMPONENTS (5 pts)
(Write the final list and quantity of the components you have included in your system)

Component/part	Quantity
ESP32	1
EBOOT 30 Pieces Photoresistor Photo Light Sensitive Resistor Light Dependent Resistor 5 mm (5539)	4

Jumper Wires	In kit - need plenty!
USB-C to USB-C cable	1
Breadboard	2
5050 Cool White LED w/ Integrated Driver Chip - 10 Pack (~6000K) [LED used to simulate lamp lightbulb]	1
Bojack 1000 pieces Carbon Film Resistor Kit	4
Push Buttons (from lab kits)	2
5050 LED breakout PCB - 10 pack![ID:1762] [used to solder LED onto for easier usage]	

PROJECT IMPLEMENTATION (30 PTS)
Tasks/Milestones Completed (15 pts)
(Describe the main tasks that you have completed in this project. Max 250 words).

Task Completed	Team Member
INITIAL MILESTONES: Week 8: Proposal Submitted Week 9: Proposal Either Accepted/Revised Week 10: Set Up the Hardware & Photoresistor Testing - Set up photoresistors, TTGO - Tested photoresistor functionality to measure light levels accurately Week 11: Implemented the Light Level Reading and Controlling Functions - Able to determine average light levels during the day, thresholds to use, etc. - Initial implementation used mini LEDs from lab kit Week 12: Configure to Wifi and Connect it to AWS Cloud Server (Submit Progress Report) - Set up our cloud server and worked on our server.py file, adding additional functionality to our project - Began deciding what we wanted our user interface to display and what functionality we wanted our users to have with the product Week 13: Improving Light Functionality & Cloud Server Details (finalize Dashboard, etc.)	All tasks in the milestones were completed together with both members.

- Implemented the new LED light, worked on and finalized Dashboard site - Implemented the button functionality, hooking it up to our system (waited off on this a little bit to be able to debug other simpler system issues before messing with the implementation of the button Week 14: Make Progress Checkin Videos - Debugged as we needed for better video quality Week 15: Final Testing and debugging of system Week 16: Project Presentations (Report, Demo, and Code due)	

Challenges/Roadblocks (5 pts)
(Describe the challenges that you have faced and how you solved them if that is the case. Max 300 words).

Originally, were going to use a standard lamp and use an LED Dimmable Lightbulb. However, there were complications since it would be a safety hazard and voltage compatibility did not allow for this implementation. Another challenge was how we moved on to using 4 mini LEDs to replace the LED Dimmable Lightbulb. Since the implementation is not large-scale (or usable in a real-world environment), we had to create a simulation using a box, and had the 4 LEDs simulate being a bright light source. The 4 mini LEDs were not producing enough light to make noticeable changes in photoresistor readings or light level in the room, so we finally settled on using the Adafruit LED strip (1 LED), since the individual LEDs in these strips are very, very bright and resemble more of a real-world lightbulb due to the amount of light they can produce. After implementing this new LED, we did not face any major challenges. Another slight challenge we had was determining what the user interface would look like. Not having much skill in apps or website, we chose the simple html site route.

Tasks Not Completed (5 pts)
(Describe the tasks that you originally planned to complete but were not completed. If all tasks were completed, state so. Max 250 words).

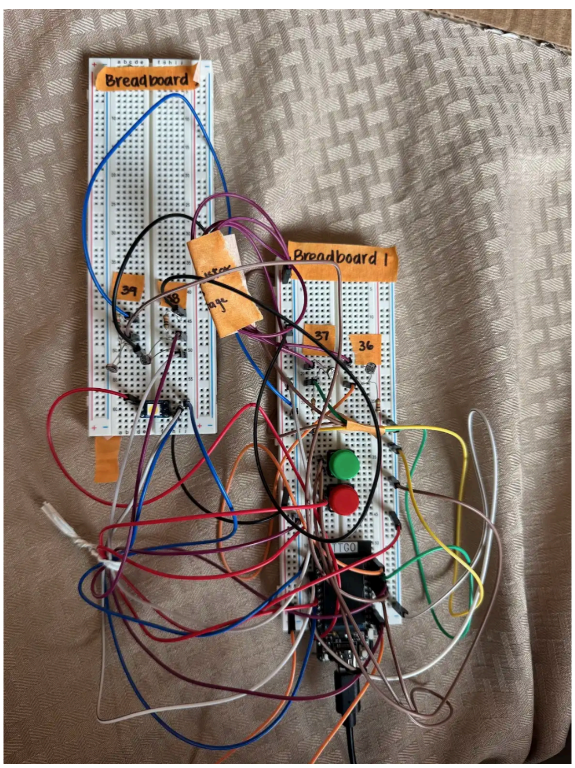
Task	Reason
None	**All tasks were completed.**

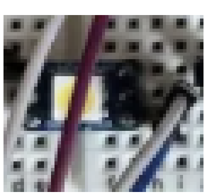
WEAK POINTS / FUTURE WORK (15 pts)
(Mention at least two points of your project that have room for improvement. These points can be additions to the existing project setup (components) or improvement of the current implementation. Max 200 words).

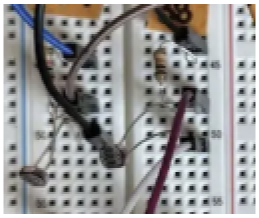
In the future we would like to add some more advanced functionality. First, we would like to store the user's light preferences. Another thing we would like to add is being able to set the standard light level to the appropriate light level depending on the time of day. Since these are both more advanced features, this is something that a downloadable application would do, allowing for more user input and functionality (or even just a more advanced and fancy website). Due to the nature of time to complete the project and lack of UI experience, creating a whole application to incorporate these features was not reasonable. Lastly, we would want to evolve this project into using an actual standard LED lightbulb and lamp, which takes more knowledge in power and electrical engineering.

SOURCE CODE (25 pts)
Please include a link to the source code of your project. A link to a repository (like GitHub) is preferred.

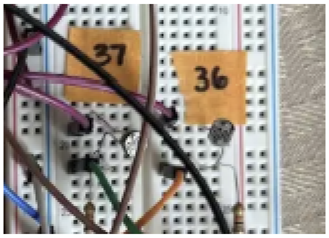
https://github.com/amom1053/CS-596-Project




LED



Photoresistors - 38 and 39



Photoresistors - 36 and 37