Efficient Light Control System for Plant Growth

By

Christelle Seri (seri2),

Heonjang Lee (hl8),

Sungjoo Chung (sungjoo2)

ECE 445 Senior Design Project Proposal

Fall 2022

TA: Zhicong Fan

09/15/2022

Project 5

Contents

1. Introduction

1.1. Problem

Greenhouses in the industry are essential in the agricultural fields, but also costly. Greenhouses are proven to be an effective solution to growing plants, but over time, the electricity costs will begin to add up.

1.2. Solution

This project proposes an energy efficient blind system with UVA lights as a solution to simulate a cost effective greenhouse system. A sensor would be placed on the plant vase to measure the amount of light received. The blinds would adjust so as to optimize the amount of light to the plant. The UVA lights will turn on when the maximum sunlight from the blinds is insufficient.

Thus the UVA lights would only be used when strictly necessary, cutting down on electricity costs as a result. Additionally, the blind system could be scheduled and adjusted to user needs as well.

This system will be easily controlled by a user using a mobile application, and also statistics will be provided on the application.

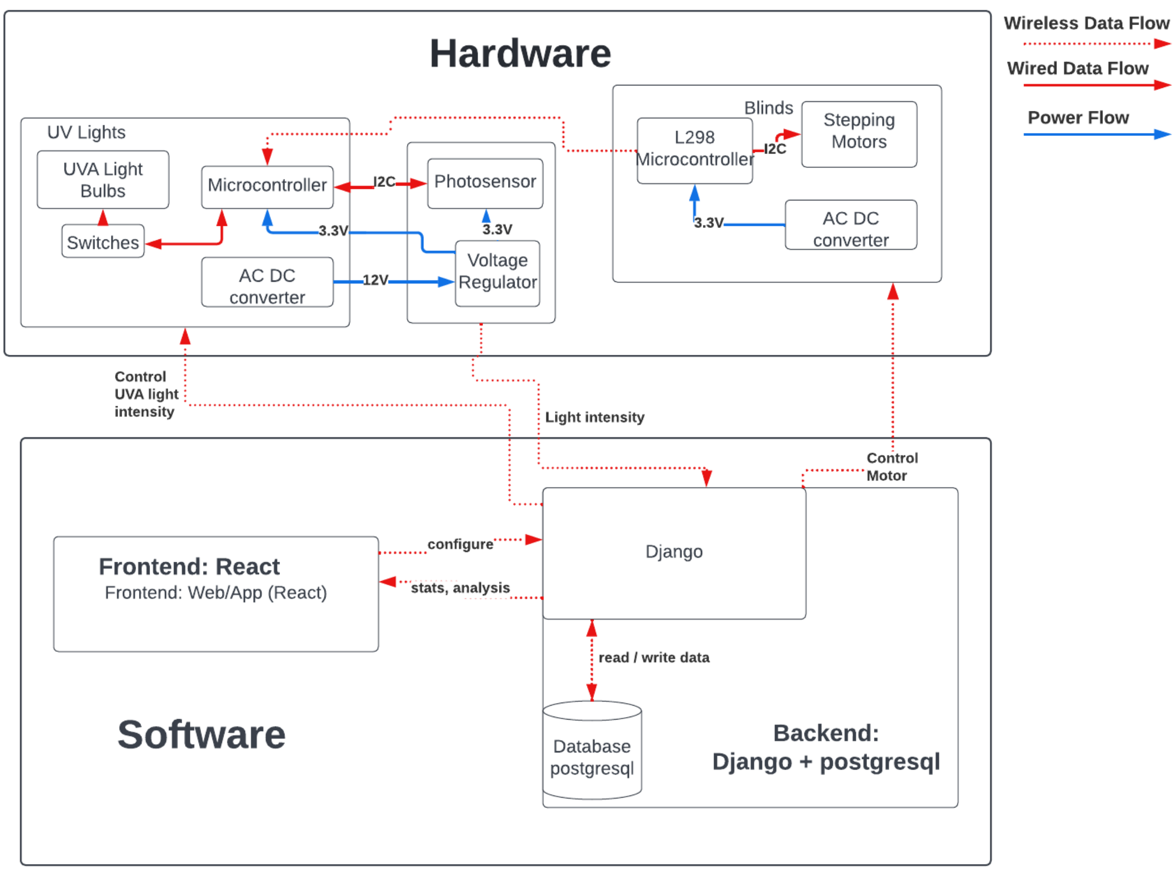
1.3. Visual Aid

1.4. High Level Requirements

- Regardless of the weather conditions, the system should be offering a constant amount of light for a requested amount of time.
- The photosensors on the vase should correctly calculate the illumination on the plant to minimize the discrepancy between the actual illumination on the plant and the expected illumination
- The application should have enough modes to cover various types of plants including cactus, tropical plants, conifers, etc

2. Design

2.1. Block Diagram



2.2. Subsystem Overview

Photosensor

The photosensor used will be a TSL2561 Luminosity sensor. The advantage of this sensor is that it is a precise digital device, and compact in size as well. There will be an ESP32 which will communicate directly with the TSL2561 via I2C. Luminosity data will be transmitted to the app through bluetooth.

Requirement 1: The photosensor subsystem will transmit real time luminosity data to the rest of the system via bluetooth

Requirement 2: The photosensor's measurements will reflect changes to the light to the plant

UV Light

To maximize the energy efficiency of the system, an adjustable UVA light circuit will be implemented. Mainly, there will be multiple 5W light bulbs. As increasing UVA light is needed, increasing switches will be flipped on to power more UVA lights.

There will be an AC-DC power supply to power the subsystem. An LM2651 will be used to step down the voltage to 3.3 V for the microcontroller. The ESP32 like in the other subsystems will send data points over bluetooth

Requirement 1: The UVA lights should be able to adjust to either increase or decrease the light to the plant

Requirement 2: The UVA light subsystem will transmit real time data on the power used

Motorized Blinds

For the motorized blinds, we will be using the Grove L298P I2C Motor Driver. Two important factors taken into consideration is the ability to communicate via I2C and feedback capability of the servo motors on the chip. Servo motor is crucial because it provides positional feedback to our controller to make further modifications. This subsystem will also communicate over bluetooth.

Requirement 1: The motors should provide enough strength to control the blinds

Requirement 2: The motors should be able to rotate in both directions

Requirement 3: The motors should be able to provide position feedback if it is required

Phone Web Application

To let a user control and monitor the entire system, a phone web application will be built. The phone application consists of a backend and a frontend.

1. Frontend

   - This part is where the user makes an interaction with the system. Users will enter the mode they want to run in the system, and this frontend will deliver that configuration to the backend. Also, the statistics/analysis of the system will be passed from backend to frontend and will be shown to the user in a user-friendly way.

   - Requirements:
     a. The frontend server should be running in the AWS consistently so that the user faces no inconvenience.
     b. The system should have < 500 ms of latency.

2. Backend

   - This part is where all business logic happens. Light intensity data points will be passed from the photosensor subsystem and then stored here, and the system's configuration will be passed from the frontend. Using those two data, it will send commands to the UV light and Motorized Blinds subsystems. Also, it will periodically (tentatively every end of the date) summarize the use, and report the data to the Frontend.

   - Requirements
     a. The backend server should also be running in AWS reliably so that the system does not die. This is critical in the system because having the system fail can harm the plants in the system. Therefore, in order to strengthen the reliability of the backend, k8s will be used.
     b. The system should be very reactive to changes (< 500ms latency). I.E., if there's a change in the configuration or change in the external light system, the system should immediately behave to adjust itself.
     c. There should be an internal scheduler so that the analysis is updated on time.

2.3. Tolerance Analysis

2.3.1. Hardware Component

The main risk for the hardware component to be considered is that photosensor subsystem encounters issues to sensing and transmitting data on luminosity. All other subsystems rely on real time data from the photosensor for the project to operate properly.

The biggest concern would come from the battery life of the subsystem. As this is the only battery powered part of the project, ensuring there is enough power is crucial. As such, extending the battery life as much as possible would be ideal.

The battery powers both the ESP32 and TSL2561. The TSL2651 has a minimum voltage of 2.7V, a nominal voltage of 3V and a maximum voltage of 3.6V. The ESP32 has a minimum voltage of 3V, a nominal voltage of 3.3V and a maximum voltage of 3.6V. We will perform a comparison of the expected minimum and maximum battery life when we operate these two subsystems at their combined minimum and maximum voltages. We will be using 2 AA / AAA batteries connected in series in order to provide a nominal voltage of 3V and a capacity of 2000mAh. We will be using the equation below to calculate the total battery life in hours.

$$H = \frac{I_{total} * Vcc}{C}$$

Table 1: Total Battery Life Calculation with ESP32 and TSL2561

	Current Drawn (mA)				
Voltage (V)	ESP32	TSL2561	Total Current (mA)	Total Power (mW)	Total Battery Life (h)
3.0	500	0.24	500.24	1500.72	1.33
3.6	500	0.24	500.24	1800.864	1.11

After performing this round of power analysis, we quickly came to the realization that it is impossible to power the ESP32 with batteries. Thus, we've decided to power only the photosensor with the batteries and the ESP32 off the wall. The table below reflects the changes.

Table 2: Total Battery Life Calculation with only TSL2561

	Current Drawn (mA)		
Voltage (V)	TSL2561	Total Power (mW)	Total Battery Life (h)
2.7	0.24	0.648	3086.42
3.6	0.24	0.864	2314.81

It is clear from this round of analysis that with the battery only powering the TSL2561, we have sufficient power to operate the TSL2561 for a minimum of 96 days. We get approximately a 33% increase in battery life if the TSL2561 is powered at its minimum voltage when compared to that of maximum voltage. As the photosensor is a crucial component of our system, it's best to power it at its nominal voltage and avoid running it at its minimum voltage, even at the cost of sacrificing some battery life.

Upon further consideration, since the ESP32 is already connected to the TSL2561 for transmission of data, it would be much more convenient to power the TSL2561 with the ESP32. When checking the data sheet, the ESP32 provides a $V_{OH} = V_{DD} * 0.8$. If we can guarantee a $V_{OH} > 2.7V$, we would be able to power the photosensor without concern about battery life at all.

2.3.2.

In the software aspect of things, the system should have some level of authentication to protect it from the hackers. If this system was used at an industry level, failing to maintain a desirable environment would seriously damage the plants under this system. Also, because system failure is critical, the backend server should be able to recover itself from unexpected failures.

3. Ethics and Safety

This project is subject to [1]the ACM Code of Ethics 1.3*Be honest and trustworthy*. Because we are integrating all existing technologies into one system, we are destined to borrow ideas or approaches made by other people. There, we should cite those ideas properly to recognize the original author. Also, privacy should be taken seriously which is related to [2]the ACM Code of Ethics 1.6. Our system stores user's authentication and they use histories in our database. This should be most firmly protected to avoid any privacy leakage.

This project is also subject to potential fire accidents due to UVA light systems at an industry level. Both UVA lights, motors, powers should never be overloaded to comply with [3] the ACM Code of Ethics 1.2 *Avoid harm.*

References:

[1], [2], [3] https://www.acm.org/code-of-ethics