KnightTint
Shade, Safety, Style Combined

EEL 4914 | Senior Design I | Spring 2024 | Group 8

Emmanuel Levasseur
Computer Engineering
University of Central Florida
em323965@ucf.edu

Luckner Ablard
Computer Engineering
University of Central Florida
Luckner.Ablard@ucf.edu

Oren Muszkal
Electrical Engineering
University of Central Florida
Oren.Muszkal@ucf.edu

Stephen Polner
Electrical Engineering
University of Central Florida
st628169@ucf.edu

Committee:
Dr. Abichar
Dr. Wasfy
Dr. Mike
Advisor:
Dr. Chan

Table Of Contents

1. Executive Summary

For thousands of years, windows have been built into our structures to allow light and give us visibility to the outdoors. When glass became readily available, people integrated it to allow their homes to be sealed from the outside while maintaining this visibility. However, when artificial lights became a mainstay in our society, we ran into a new problem: light pollution. People began trying to find solutions for the mass amount of light entering their rooms, which was heating the room up and disrupting their sleep. Curtains became the go-to solution for this problem and also provided privacy from the outside world, eventually resulting in the development of blackout curtains, which completely blocked out the outside light. However, modern style choices have moved away from these traditional curtains, and thus, people have been designing solutions fitting the minimalist lifestyle. The idea of modifying glass with special films arose as a solution. By controlling these special films, people have been able to find a balance between lighting, privacy, and protection from solar radiation.

Our project aims to leverage this new innovation with pre-existing technology. This system will use this data to automatically adjust the tint of the windows in a convenient manner that allows them to save the user the money on the HVAC electricity costs associated with cooling their rooms. KnightTint is designed with the intent to provide consumers the freedom to allow the system to tint their windows to ensure the temperature control of their homes while allowing them to control their privacy from the outside world. To facilitate this, our tinting system will use a temperature sensor and light sensor to accurately measure the amount of light entering their rooms and the temperature of their rooms. These two sensors will be on or connected to a custom PCB which we will design to house an MCU that will communicate with a centralized microcontroller board. This MCU will be in charge of doing all of the algorithmic work for controlling our windows and making any adjustments to our windows.

This report outlines the development of KnightTint, beginning with the initial inspiration, objectives, and aims of the undertaking. It then transitions into a thorough examination of the specific requirements and the strategic approach our team adopted to construct our system. The third chapter, dedicated to technological exploration, elaborates on the rationale behind the selection of each component and the progression to those choices. The subsequent section delves into the challenges encountered, such as managing power efficiency and user-friendliness, along with the industry standards that guided our project's development. Following this, the document delves into the intricate design elements, offering diagrams and detailed descriptions of the hardware, software, interfaces, and the system's blueprint. The narrative then shifts to the integration of the bespoke PCB within our intelligent window tinter, detailing the PCB's design and the testing protocols applied. The report concludes with a section on project management before summarizing the project's outcomes and learnings.

2. Project Description

2.1 Introduction

The purpose of this project is to increase convenience, provide privacy, and address the escalating concerns over energy consumption, particularly in HVAC systems, within the context of a world marked by continuous development and population growth. By collecting and analyzing temperature and light intensity data, the system aims to dynamically adjust window tint percentages, matching user preferences and contributing to energy efficiency and a sustainable lifestyle by blocking solar radiation. As for the stretch goals, the project envisions implementing presence detection at each window to increase transparency when someone approaches to look outside. Artificial Intelligence is proposed to adapt the system's automation based on user activity and preferences, adding a layer of sophistication to the overall user experience. Ultimately, the project seeks to offer a holistic solution that not only addresses energy challenges but also aligns with the growing trend of smart home technology.

2.2 Motivation and Background

In a world marked by continuous development and population growth, alongside significant advancements in science and technology that contribute to increased life expectancy, the issue of energy consumption has become a growing concern. The escalating awareness of our environmental impact and the strain on limited resources has prompted an urgent call to action. It is imperative for us to acknowledge and address this problem by incorporating numerous energy-saving systems in our surroundings.

When examining our homes and the various devices contributing to energy consumption, HVAC systems emerge as the most energy-intensive. According to the Department of Energy, heating and cooling account for approximately half of a typical home's energy usage. Effectively addressing this challenge requires a concerted effort to assist HVAC systems in minimizing their energy consumption. By tackling the intense energy use of these systems, we not only contribute to a more environmentally friendly future but also deliver tangible benefits to homeowners, including potential cost savings and increased energy efficiency. As Floridians, we understand firsthand the energy cost challenges, particularly during the summertime, and the importance of supporting HVAC systems for a more sustainable lifestyle.

The concept of the smart tinting window system originated from a team member who encountered Smart Film PDLC. PDLC is a film that rapidly transitions from opaque to transparent when a specific amount of electricity is applied, commonly used in commercial spaces with a binary on/off system. It can block a significant portion of infrared light, thus reducing solar heating. Our idea was to introduce a system that utilizes this (or an alternative) smart glass technology to residential customers, capitalizing on the growing trend of smart home technology. As individuals increasingly seek to connect and automate every aspect of their homes, this system can facilitate and enhance those connections.

Another team member observed the potential use of this film in other automated systems, particularly in cars. Recognizing the almost essential need for window tinting in states like

Florida, where dealing with intense heat and sunlight is challenging, an automated shading system for car windows could mitigate heat-related incidents during hot Florida summers. We considered adding a detection feature to identify when a person is in the car, activating the shading accordingly. Additionally, integrating the system into the car's computer would allow for automated control of the shading. The decision to focus on the house system was influenced by complex legal and safety considerations surrounding the car window system. Ensuring compliance with state laws for the car system would demand meticulous attention to detail. We also took into account potential risks to customers' cars, avoiding interference with the complex computer systems and recognizing the associated challenges.

2.3 Existing Product / Past Project / Prior Related Work

Numerous commercial companies provide smart glass systems utilizing the PDLC system for window shading. This technology proves versatile in settings like office spaces, where shades can be activated during meetings or privacy levels controlled based on activities within the building. These systems are extensively used in work offices, catering to diverse needs.

One noteworthy company offering smart window film systems in our area is Gauzy. Primarily marketed for privacy applications, Gauzy showcases examples in office spaces where occupants can seamlessly toggle the system based on room activities. Gauzy has improved the glass, adding a scratch-proof coating for durability. The adhesive used ensures longevity, preventing peeling over extended use. Their system features default tinted glass, clearing only when an appropriate amount of electricity is applied. Gauzy offers programmable controllers for their glass, but does not design automated control systems.

Another player in this space is Smartglass Technologies, providing similar services to Gauzy but with a broader range. While emphasizing commercial office spaces, they offer additional services, including an online-connected interface through an app. Smartglass Technologies targets a wider audience, particularly focusing on the healthcare sector, promoting privacy solutions in areas like surgery rooms. However, a common issue among these companies is the dependence on their services, tying users to their apps and subscriptions for the devices to function. This reliance on subscription services, amid a trend towards monthly subscriptions, may pose potential issues later on. Our goal is to provide an independent product, granting users complete control without hidden features behind paywalls.

Despite the prevalent use of this film in various systems, there's a concentrated emphasis on commercial spaces like offices and meeting rooms. While valuable, there is untapped potential for energy savings. Implementing the same film used in offices on windows, coupled with a system that detects sunlight levels to decide whether to tint or not, could significantly contribute to energy efficiency. Furthermore, integrating the window with the HVAC system could optimize temperature control, utilizing sunlight to either cool down or warm up the space based on user preferences.

2.4 Goals and Objectives

The primary goals of this project are to offer the satisfaction and convenience of an automated window tinting system, including the maintenance of a desired indoor temperature and light intensity, as well as to save the user money on electricity costs associated with indoor cooling.

This will be done by integrating the existing technology of smart glass with a uniquely designed control system. Our control system will be implemented in a building - a model house for the scope of this project, and it shall adjust the tint percentage of windows based on indoor temperature and sunlight intensity shining on the building's windows. The advantage of this system over a static tinting system is its ability to adapt to changing conditions and maximize outdoor visibility when tinting is not appropriate for the current conditions.

On the software side of this project, there is an essential goal to design a mobile application that allows users to manually control the tinting system and customize functionality to their liking.

2.4.1 Basic Objectives:

These objectives are critical for the functionality of our project.

1. Collect temperature and light intensity data using thermal and luminosity sensors.

In order to achieve the goal of targeting the user's desired light intensity and indoor temperature, we will of course need sensors to determine the present values of these factors. This information will be essential for our automated system's tinting decisions. We will choose and place sensors to ensure accuracy and reliability, avoiding user frustration. The thermal sensors will be placed in a discreet location on the interior wall to measure the indoor temperature, and the luminosity sensor will be placed on the exterior wall, as we want to measure the light before it goes through the tint.

2. Transmit data via wireless relays to and from a central microcontroller.

This system will have a central controller that will control the tint percentage of all of the windows. The data from the sensors must be transmitted to this unit, and the instructions from the control must be transmitted back to the window. However, the controller cannot simultaneously be close to all of the windows of the building. To solve this, we will implement wireless relays to mitigate the need to run an unsightly wire from each window to the central controller or hide said wire. This solution helps achieve the goals of providing user convenience and satisfaction, as this makes the system easier to install, less cumbersome, and more aesthetic. Additionally, these wireless relays will integrate with the user's home Wi-Fi network, allowing the user to interface with the device over a network they already understand and facilitating smartphone app control.

3. Program microcontroller logic to output voltages that act as a control signal.

Once sensor data is transmitted via wireless relays to the central control unit, the microcontroller inside of it will compare sensor data to user's desired indoor temperature and light intensity, then determine if tinting the windows will aid in cooling. It will increase tint percentage if it is both warm inside and sunny out, or if the light intensity surpasses the desired level. The output voltage will be a wireless control signal and does not power the tint itself. The logical operations carried out by the microcontroller are essential for achieving the goal of making this a dynamic system that adapts to changing conditions. This helps increase electricity savings while also maximizing outdoor visibility when possible.

4. Design a PCB and implement a controller to transform control signals into the proper voltages for the desired tinting percentages.

In order to actually change the tinting percentage at each of the windows and achieve the basic functionality of dynamic tint outlined in our goals, we will need to transform the wireless instructions from the central control unit into the proper voltages to change the tinting percentage. We will integrate a dimmable controller in order to output the higher voltage necessary to tint the windows. There will be a PCB that contains a wireless transceiver and a basic microcontroller to transform these signals into the appropriate control voltages for the dimmable controller. The PCB and dimmable controller will be as small as possible and discreetly placed in order to reduce the footprint of the system at the windows and avoid creating an eyesore, which would lower user satisfaction.

5. Implement an AC to DC converter to power the central control unit.

We will power the central control unit hardware from an outlet, using a voltage converter to allow it to be powered easily via USB. We do not want the central unit to die at any point, or all of the users would lose control over all of their windows, which would be very frustrating and run counter to our goal of user satisfaction.

6. Incorporate a smartphone application into the logic of the microcontroller.

We do not want the user's only mode of control over the system to be determined by the automation outlined above. The user may desire a different tinting state than that determined by the automated logic, and it would be very frustrating to lack control over the system's output. We will design a smartphone application that will give the user increased control and customizability over the functionality of the tinting system. In the modern age, virtually everyone has a smartphone, so using this technology is very convenient and familiar to the user. Basic app features will be setting desired indoor temperature and light intensity values, a manual tint percentage override and viewing of sensor data.

2.4.2 Advanced Objectives

These objectives are more complex and expand or improve the functionality of the system.

1. Minimize response time.

The system would not meet its goal of increasing convenience if there is a noticeable delay after conditions change or the user performs an action on the app. We will design the relay system and optimize the microcontroller code to reduce latency. This will help mitigate user frustration.

2. Design and incorporate advanced app functionality.

We want the user to have control of the system beyond simple manual control. The app will have more advanced features, including allowing users to control each wall of windows individually and create a schedule for the windows' tint levels. They will be able to toggle between automatic, manual, and scheduled control. This increased level of control will help increase user convenience and satisfaction by creating customizable satisfaction.

2.4.3 Stretch Objectives

These objectives are not essential for the core goals project, but would add additional functionality. These objectives will only be worked towards only once all other objectives have been completed and may fall outside the scope of the project.

1. Integrate with Smart-Home technology (Alexa and Google Home/Nest).

User convenience is an essential goal in this project and we recognize that many other smart-home products integrate with Alexa and Google Home/Nest, the two most common smart-home controllers to achieve this. Doing so would enable voice commands to control the tinting system. This would add another avenue of control beyond the automation and app controls, which further increase convenience and accessibility. An example command could be, "Alexa, darken the windows in the living room".

2. Implement fail-safe mechanisms.

Inevitably some part of the system will fail, and we want to make sure that the system has a stable state as its default. If a sensor malfunctions at a window, we will utilize data from the closest sensor, on the same wall if possible, to control that window's logic. In the event of a microcontroller or WiFi failure, we will design the system to revert the windows to a default tint percentage. We do not want the windows to become completely opaque or fluctuate widely, which could cause user frustration and detract from our goal of user satisfaction.

3. Implement presence detection sensors.

In the current scope of the project, the automatic tinting of windows may create an undesired limitation of outdoor visibility, and the user would need to go on their smartphone application to perform a manual override. This is an inconvenience that could be avoided if the window automatically untinted when approached. We would place presence detection sensors on the interior of each window and program the microcontroller to lower the tint percentage at each window when someone approaches to view outside. This solution would expand functionality towards our goal of maximizing user convenience.

4. Implement UV sensors.

A common reason people install tinted windows on their building is to increase protection from ultraviolet rays. In order to expand the functionality of our system to protect from UV rays, we would place UV sensors on the exterior of each window and program the microcontroller to darken the windows when the ultraviolet light coming through the window is dangerously high, which can occur even when it is not bright outside. This would increase user safety and health, which could increase user satisfaction.

5. Utilize live weather data.

Instead of expanding the hardware scope with additional UV sensors, we could pull UV index data from a weather database and adjust the tint percentage accordingly for user safety and health for similar user benefits. The live weather data could also be used to control the tinting system more accurately and potentially increase electricity savings from cooling.

6. Incorporate artificial intelligence.

The ultimate step towards our goal of user convenience would be a system that intelligently adapts to a user's preferences. The system would seek out patterns in the smartphone app interactions and learn their daily work and sleep schedules. The AI would dynamically regulate the tinting of the windows to provide an enhanced and personalized user experience.

2.5 Requirements and Specifications

This section outlines the key features of the project, as well as details on the hardware and software requirements that will enable them to function.

2.5.1 General

- A window tinting system that can be set up in different modes to tint to different levels based on the user's need and specific specifications.
- The windows can always be controlled manually through the app so that the user can control it manually if needed.
- Temperature and light sensors that take in the current environment reading and adjust the tinting level based on the current amount of light and indoor temperature.

● Minimum 1 pair of windows connected and communicating with one another.

2.5.2 Hardware

● A light sensor on the outside of the window takes in information from the outside and sends input to the MCU.
● A temperature sensor to read the current temperature of the room.
● At least two windows of smart glass to allow users to shade the window at different levels based on the user's needs.
● A custom PCB for the individual window with the sensor and power control for the tinting system.
● A custom PCB for the control station allows for the sync and control of the windows.

2.5.3 Embedded Software

● Integration of sensors and devices
  ○ The MCU will get information from the sensors and adjust the tinting values accordingly.
  ○ Inclusion of different modes allowing the user control over the system.
  ○ Wireless communication between the windows and the central MCU for seamless integration.

2.5.4 App and UI integration

● Dashboard Access:
  ○ Quick access buttons for common actions like "Tint All," "Untint All," and customization options.
● Scheduling Features:
  ○ Flexible scheduling options for automated tinting based on time, day, or external conditions.
  ○ Customizable schedules for different windows or groups, enhancing user convenience.
● Push Notifications:
  ○ Instant push notifications for essential events, including active tinting or system malfunctions.
  ○ Keeps users informed and in control of the smart tinting system.
● User-Friendly UI Design:
  ○ Visual representation of the house layout with interactive icons displaying real-time window tinting status.

2.5.5 WiFi Connectivity

● Connecting the individual windows allows them to sync when it comes to the sensor reading.
● Connect the user with the System allowing them to control from the applications.

2.5.6 Power

- Windows will be connected to an MCU with a dimmable controller and will be connected with an AC power supply.
- The MCU PCB will be powered by an AC power supply.

2.6 Marketing and Engineering Requirements

The table below describes the various engineering and marketing requirements that are relevant to our project. The three columns outlining the engineering requirements we have laid out for our project and the aligned marketing goals. Highlighted amongst the requirements are our chosen requirements which we must provide with our device in Senior Design II.

The first of our requirements is that the sensors must read and send inputs every second. Most sensors are capable of doing this and filtering their signals will be the valuable aspect. Frequent sensor readings will allow our system to change its behavior based on up-to-date information, ensuring more reliable performance. Our second requirement is that our service must react quickly to user demand. The user must not be waiting around for minutes on end for our system to function or else they would become frustrated with its lack of responsiveness, and the ability for the system to aid in reducing solar heat would be impaired. Our final requirement is that in less than 1 out of 100 times, our service will fail. This means that our service will nearly always be reliable, which is essential to ensure the user experience of our product is satisfactory.

Marketing Requirements Key:

1. Ensure rapid system performance.
2. Delivering quality and luxury while maintaining affordability.
3. Ensure reliability in temperature and light sensing.
4. Keep device maintenance requirements low.
5. Ensure accessibility for all mobile users.
6. Design a user-friendly interface.
7. Make user experience pleasant and convenient.
8. Install the system in an aesthetic, non-obstructive manner.
9. Ensure the system aids the user's air conditioning system in keeping the building cool.

Marketing Requirements	Engineering Requirements	Justification
2, 4, 7	The MCU should have enough power to run for 1 month.	The system should have enough battery life to be "hands off".
1, 3, 9	The sensors should read the temperature and light at least once per second.	The system should be adaptive and respond to rapidly changing circumstances.
2	The cost of the device should not exceed $500	The device should be a luxury but

	with some of the tinting.	somewhat affordable.
1, 7, 9	Real-time Tinting Updates: Updates with < 5 seconds delay.	Timely updates enhance user satisfaction by providing real-time information about window tinting.
1, 6	Achieve an average UI response time below 300ms.	Ensuring quick and smooth response to user interactions within the mobile application.
7	Connectivity Stability: < 1% occurrence of connectivity issues.	Ensuring stable connectivity between the mobile app and smart tinting devices.
8	Sensor dimensions should not exceed 3" x 3" and PCB dimensions should not exceed 5" x 5".	This system should increase convenience and aesthetics, not create an eyesore.
7, 9	MCU logic should accurately control tint based on sensor data to meet user preferences and save money on cooling. <1% of inaccurate behavior.	A system called "IntelliTint" should display intelligent behavior.
9	System should aid in reducing indoor warming from the sun and reduce cooling electricity usage. >10% electricity savings.	This system should be an investment by the user and save money in the long-term.
5, 7	Wireless capability range between MCU and window PCB, and MCU and smartphone, should be at least 100'.	The MCU should be able to communicate with any window or phone in a standard-sized house.

2.7 House of Quality

The table below depicts the trade-off relationships between the customer and engineering requirements of this project. It also depicts our target trajectory for each quality, the ranked importance of each customer requirement, and the target values for the engineering requirements. The "roof" of the house shows the trade-off relationships between the engineering requirements. This is a visual aid for our group to use when making decisions on how to allocate funding for this project, as well as focus efforts to improve the different aspects of our system's technical functionality in order to meet customer requirements.

House of Quality Legend	
▲	Maximize
▼	Minimize
↑	Positive Correlation
↑↑	Strong Position Correlation
↓	Negative Correlation
↓↓	Strong Negative Correlation

House of Quality Table:

Importance Ranking	Direction of Improvement	Customer Requirements / Engineering Requirements	Battery Life	Frequent Sensor Polling	Cost of Components	Tint Update Delay	UI Response Delay	App Connection Stability	Hardware Dimensions	Electricity Savings	Accurate Tint Control by MCU	Wireless Range
		Direction of Improvement	▲	▲	▼	▼	▼	▲	▼	▲	▲	▲
3	▲	Performance Speed		↑	↑	↓↓	↓↓					
9	▲	Luxury	↑		↑↑	↑	↑		↓	↓	↑	↑
4	▲	Sensor Reliability		↑↑	↑	↓					↑↑	
7	▼	Maintenance Requirements	↓↓		↓						↓	
5	▲	Accessibility for All Mobile Users					↓	↑↑				↑↑
6	▲	User-Friendly App Interface					↓↓	↑				
1	▲	Pleasant/Convenient User Experience	↑		↑	↓↓	↓↓	↑↑	↓		↑↑	↑↑
8	▲	Aesthetics			↑				↓↓			↑
2	▲	Indoor Cooling		↑	↑	↓	↓			↑↑	↑↑	
Target for Engineering Requirements			> 1 Month	> Once Per Second	< $500	< 5 Seconds	< 300 ms	< 1% of Connectivity Issues	Sensors: < 3" x 3" PCB: <5" x 5"	>10% Bill Reduction	< 1% Innacurate Behavior	>100'

2.8 Hardware Block Diagram

The block diagram below gives a very high-level overview of the hardware. This diagram consists of our central MCU development board, our two window PCBs that we will include in our demo, and the window boards' associated tinting hardware. We also show that the control hardware for our system will be a smartphone, which is of course not designed by us. On this diagram, solid lines will be wired connections while the dotted lines represent a wireless connection.

The development board has many components, but we are not designing this board, so we have included it as a single block. However, the booster pack for our development board is shown, as this is a separate component that easily attaches to the development board and extends its functionality. The function of the development board is to act as a central control center for our window boards.

The significant PCB design for KnightTint comes in the form of these window boards. The function of these boards is to receive data from our sensors and relay it to our development board, as well as relay instructions from the central MCU to the dimmable controller, thus controlling the tint. On the PCB, we have a temperature sensor, which will measure the temperature of the room, a protocol converter that allows for connection to the dimmable controller, a Wi-Fi module, which will allow for wireless communication, a battery compartment, which will hold the batteries that will power each window MCU, and a voltage regulator, which will step down this voltage to a constant 3.3V, which is the normal voltage supply for microcontrollers and many sensors. The light sensor is shown external to the MCU, as it will be mounted outside of the window. The dimmable controller is powered by an outlet, as it has to output a high AC voltage to the tint.

The work distribution is shown in the legend below the block diagram. The majority of hardware responsibilities have been assigned to Stephen Polner and Oren Muszkal, leveraging their extensive hardware experience. Emmanuel is tasked with the MCU block, responsible for deciding on the MCU and overseeing its programming. Luckner Ablard is spearheading the development of the user-side software application, which is why his responsibilities on this diagram are limited.

Hardware Block Diagram
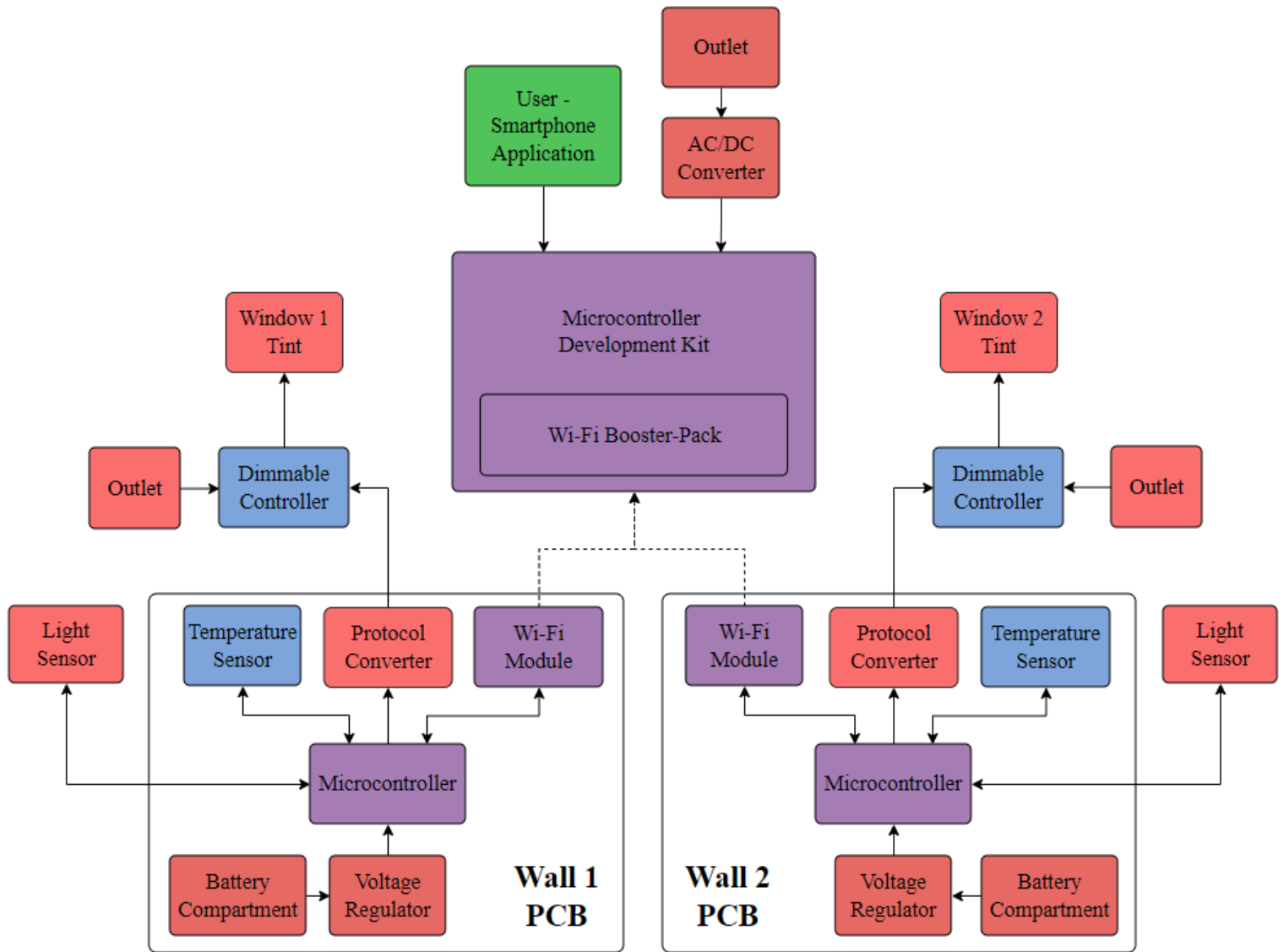
2.9 Software Flowchart

The provided diagram illustrates key software aspects integral to our project. To realize the desired features, we envision implementing a mode system, enabling users to switch between different operational modes. The initial focus lies on implementing fundamental modes, such as user-controlled mode utilizing analog input through knobs and buttons. This is deemed essential as, despite perfect error handling within the software, unforeseen situations may arise, necessitating analog control.

Consideration is also given to a privacy mode, where the tint is set to maximum regardless of the situation, only changing if the user switches to a different mode. The automated mode emerges as a central focus, leveraging the system's ability to read ambient light levels and autonomously tint the window accordingly. Integration with the house HVAC system is explored to enhance overall system efficiency.

In addition to its versatile operational modes, the mobile app plays a pivotal role in the smart tinting project by empowering users to establish customized schedules effortlessly. This functionality allows users to set predetermined times for the smart window tinting system to activate and deactivate, enhancing the overall user experience. For example, individuals may utilize the app to schedule window tinting until sunrise, ensuring a gradual exposure to natural sunlight upon waking up. This purposeful integration of scheduling capabilities not only contributes to energy efficiency but also aligns with users' preferences and daily routines. The mobile app serves as a centralized hub, offering a user-friendly interface for seamless control, scheduling, and personalized customization, making the smart tinting house project a truly intelligent and adaptable solution for modern living environments.

Software Flow Chart
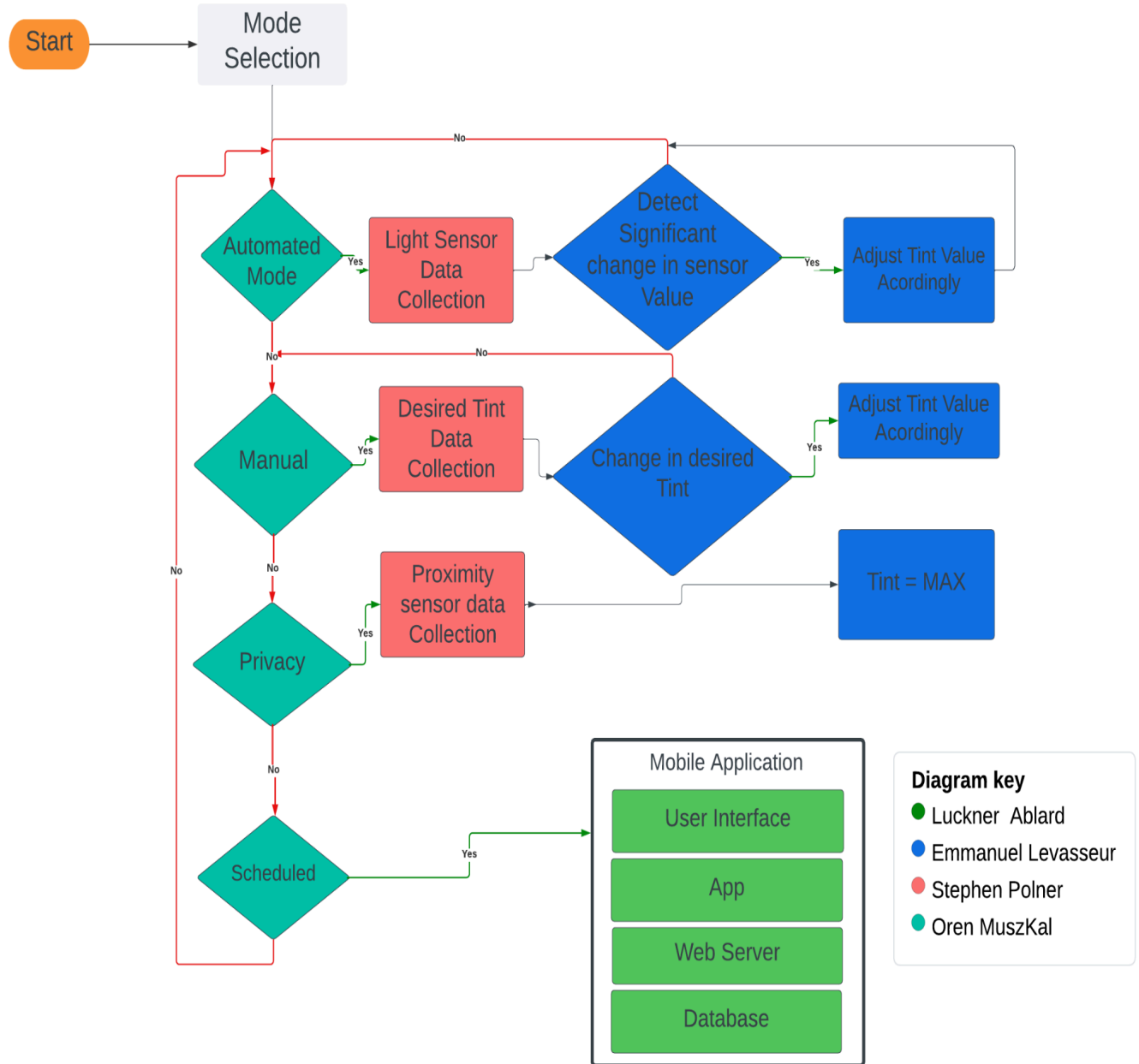
3. Research

3.1 Smart Glass

3.1.1 Smart Glass Technology Comparison

Thermochromic and Photochromic Glass

Thermochromic and photochromic glass change their transparency based on temperature variations and UV light exposure respectively. These types of smart glass are often used for passive temperature and light control. Unlike other types of switchable glass, they do not respond to applied voltages. This passive operation makes thermochromic and photochromic glass particularly beneficial in sustainable building designs, helping to reduce energy consumption by optimizing natural light usage and improving thermal efficiencies without the need for electrical control systems.

Electrochromic Glass

Electrochromic is a technology that uses an applied voltage to switch between dark and light glass. The glass typically consists of five layers: two conductive coatings, two electrochromic layers, and an ion-conducting separator in the middle. Its natural state is transparent, but when a voltage is applied, ions move from one electrochromic layer to the other, causing the glass to darken. However, the glass requires little electricity to maintain its tint. The tinting level can be precisely controlled by varying the voltage, allowing for adjustments from clear to dark. At its most translucent, 60-80% of visible light can typically pass through, and at its most opaque 5-10% of visible light can typically pass through. This process is relatively slow compared to other smart glass technologies, often taking several minutes to transition fully. Performance can worsen over time.

Dimmable PDLC (Polymer Dispersed Liquid Crystal) Glass

PDLC is another switchable glass technology that is controlled by electricity. PDLC glass consists of a layer of liquid crystal droplets dispersed within a polymer matrix. When an electric current is off, the liquid crystals are randomly oriented, scattering light and making the glass appear opaque (frosted). When electricity is applied, the liquid crystals align, becoming transparent and allowing light to pass through. The response time of this process is very fast - often less than a second. At its most translucent, 80-90% of visible light can typically pass through PDLC glass, and at its most opaque 40-50% of visible light can typically pass through. PDLC glass usually has only two states and is primarily used for privacy, but variable opacity is possible in some cases, which would be necessary for this project. PDLC film is an option, making it very easy to retrofit existing glass.The cost of PDLC is generally low and the market availability of products utilizing this technology is high.

SPD (Suspension Particle Device) Glass

Suspension Particle Device (SPD) glass is another switchable glass technology that is controlled by electricity. SPD glass contains microscopic particles that are suspended in a fluid between two layers of glass. Its natural state is dark, but when an electric current is applied, the particles align, allowing light to pass through, making the glass clear. The response time of this process is typically a few seconds. The level of tinting can be controlled from light to dark by adjusting the voltage. At its most translucent, 50-55% of visible light can typically pass through SPD glass, while at its most opaque, 1-5% of visible light can typically pass through. Compared to other technologies, SPD glass is generally less commercially available and is not the cheapest option.

Micro-Blinds

Micro-blinds are blinds sandwiched between two panes of glass that are so small that they are practically invisible. When a voltage is applied, the electric field between the metal blinds layer and the conductive transparent layer causes the blinds to stretch and block light. This works based on mechanical alteration rather than chemical or physical changes in the material. The level of voltage can be altered to adjust the angle of the micro-blinds, allowing for multiple levels of tint. Light transmissivity range for this technology is nearly 0-100% and switching speed is within milliseconds. However, there are several factors that could make micro-blinds difficult to implement. Micro-blind-based smart glass is very hard to retrofit into existing glass windows and is usually higher maintenance than other switching glass technologies. The largest problem with this technology is that it is very new and seems to exist more within a research environment. It has nearly zero commercial availability compared to the other opinion.

Smart Glass Technology Comparison Table

	Thermo chromic	Photochromic	Electro-chromic	Dimmable PDLC	SPD	Micro-Blinds
Electrically Controllable	No	No	Yes	Yes	Yes	Yes
Switching Time	Several Minutes	<1 minute	Several Minutes	Nearly Instantly	Several Seconds	Nearly Instantly
Powered Off State	N/A	N/A	Transparent	Opaque	Opaque	Transparent
Commercial Availability	High	High	Medium	High	Medium	Low

Available as Film	Yes	Yes	No	Yes	No	No
Light Transmittance	5-90%	10-90%	5-80%	40-90%	1-55%	0-100%

Smart Glass Technology Decision

We have decided to move forward with dimmable PDLC for this project. We compared the features of each smart/switching glass technology with our marketing and engineering requirements and did further research into commercial availability for small order quantities.
The most basic feature we need our glass to support is response to active electric control. This criterion eliminates thermochromic and photochromic glass, as while these two technologies are certainly adjacent to the goals of our system, they do not have the advanced functionality that we need.

One of the core customer requirements for our system is rapid system response time. We want the system to adjust within seconds to changing conditions or user input, as a large delay would be inconvenient and frustrating for the user. Electrochromic glass, depending on the manufacturer, usually takes several minutes to change its state in response to a new electrical signal. This is too long of a delay for our system. Additionally, electrochromic glass is generally more expensive than other technologies.

Micro-blinds, PDLC, and SPD solve these issues. Micro-blinds have a very fast switching speed switching and have a transmissivity range of nearly 0-100% based on the angle of the microscopic blinds. However, the main issue with the micro-blinds technology is its availability. This seems to be an experimental technology that does not have widespread commercial availability compared to the other technologies researched. In other systems that seek to use window tint to control light and temperature, this is not an industry standard. Additionally, the mechanical nature of the blinds can cause it to have increased maintenance, which we want to keep at a minimum.

In ideal circumstances, SPD technology would be most suitable for our system. It has rapid switching speed, a large transmissivity range, relatively large industry use, and low power usage. In its tinted state, it is also dark rather than frosted, which is what one would think when imagining tinted windows. However, after doing research, we found that this technology has very few manufacturers and very restricted availability for smaller entities like our group. The minimum order sizes unfortunately make this technology unviable.

The best alternative is dimmable PDLC. While not getting as dark as SPD glass, the technology does have a wide light transmittance range. The main drawback here is that PDLC becomes frosted rather than dark, which some consumers may not prefer. However, PDLC is much more commercially available for individual consumers and is significantly cheaper than SPD. PDLC satisfies our requirements of controllability and switching time and is available from some vendors as a film.

3.1.2 Smart Glass Product Comparison

Gauzy Solar Performance LCG SmartGlass and Flex Controller

Gauzy is the industry leader for smart glass and offers multiple PLDC-based smart glass products. Their Solar Performance LCG (Light Control Glass) has a switching time of 10 ms, seconds and a minimal energy consumption average of 3w/m². These aspects will allow this glass to respond quickly to our control system while helping to save money on electricity. This glass has a transparency range of <1% while off to 50% while on. Gauzy SPD glass is low haze at 2.5% and blocks 99% of UV light while on or off. Gauzy advertises that the Solar Performance LCG blocks up to 78% of infrared rays from the sun and can reduce indoor temperatures by up to 15°C. They emphasize that their product is suitable for dynamic privacy and temperature control.

Gauzy PDLC LCG pairs with their LCG controllers, which are compatible with dry contact or DMX control. This will allow it to communicate with our control system. We would specifically choose the Flex Controller because it offers the dimming functionality that we are looking for. This is a very expensive product option, and to even make this route financially viable, we would need to purchase the controller pre-owned through a third party vendor.

Smartglass International Blackout Smart Glass

SmartGlass International is a British manufacturer of smart glass. Their Blackout Smart Glass is a dual laminate glass of both SPD and PDLC film, combining our top two technology choices into one product. This glass blocks 100% of UV rays and can block up to 99.5% of visible light. The glass offers four distinct "modes", with privacy and tint being able to be turned on or off, but dimmable control of both aspects is possible. The glass consumes 14 3w/m² and has a switching time of 10s. The combination of both technologies allows for exceptional

SmartGlass International provides the following table on the optical performance of their Blackout Smart Glass in their Technical Specifications Brochure:

	Mode 1	Mode 2	Mode 3	Mode 4
Visible Light Transmission	0.5%	0.8%	45%	26%
Clarity	0%	20%	60%	4%
UV Transmission	0%	0%	0%	0%

Mode 1 – Privacy 'Off' Solar 'Off' (Blackout)   Mode 2 – Privacy 'On' Solar Off (Light control only)
Mode 3 – Privacy 'On' Solar On' (Transparancey)   Mode 4 – Privacy 'Off' Solar On (Privacy Control only)

Because the two glass technologies operate at different voltages (70VAC and 110VAC), this product requires two different dimmable controllers that SmartGlass International sells.

Filmbase PDLC Switchable Smart Film

Filmbase is a large vendor of a wide array of smart film products. This film is adhesive and can be applied to existing glass, which makes it a better solution for existing construction. The transmittance of the gray version of this product, which we prefer, is 51-54% while on, and 25-27% when off. This product blocks a maximum of 98% of IR and 99% of UV rays, making it applicable to the energy conservation goals of KnightTint.

The film is sold with an ON-OFF switch, so we will have to do hardware modification in order to interface it with a dimmable controller.

	Gauzy Solar Performance LCG	Smartglass International Blackout Smart Glass	Filmbase PDLC Switchable Smart Film
Technology	PDLC	PDLC/SPD	PDLC
Visible Light Transmissivity (OFF)	<1%	<1%	25-27%
Visible Light Transmissivity (ON)	50%	45%	51-54%
UV Transmissivity	1%	0%	1%
IR Transmissivity	12%	Very Low - % Not Given	2%
Availability	Low	Low	High
Cost	High	High	Medium

Smart Glass Product Decision

Our first choice for PDLC was the Gauzy LCG smart glass; however, we found this option to be prohibitively expensive for our budget. We would have needed to spend upwards to $500-600 for

this product, which is too much for our self funded project. Therefore, we have decided to move forward with the Filmbase PDLC film. It will cost us approximately $140 total, which is much more affordable. Its IR blocking capability makes it suitable for this project, and the fact that its a film makes it easier to work with when constructing our prototype model.

3.1 Temperature Sensor

Temperature sensors are a key aspect to our project. They provide valuable information to the MCU by measuring the temperature inside the room and then sending that information through. In our project, we aim to use a temperature sensor inside the room next to the window so that we can get accurate readings and be able to cool the room effectively. The main parameters set by our project are size, cost, power draw, response time, accuracy, and connectivity. Our devices need to be small enough to fit on the outside of the window without obscuring the view, cost less than $20, have a low power draw to efficiently run off the MCU, have a response time below 1 second, be accurate within 1°C, and interface with our MCU well.

3.1.1 Temperature Sensor Technology Comparison

Thermocouples

Thermocouples are robust and versatile temperature sensors, relying on the Seebeck effect. The Seebeck effect is the phenomenon where a temperature difference between two dissimilar conductors or semiconductors generates an electromotive force, resulting in a thermoelectric voltage. It offers low-cost, low-power operation with fast response times and good accuracy. Its connectivity allows seamless integration with microcontrollers through dedicated signal conditioning circuits or signal conditioning ICs. Furthermore, thermocouples are highly durable and can operate effectively under extreme environmental conditions, making them suitable for a wide range of industrial and scientific applications.

RTDs (Resistance Temperature Detectors)

RTDs are temperature sensors based on the predictable change in electrical resistance with temperature. Known for high accuracy, they are commonly used in applications demanding precision. Signal conditioning is necessary for interfacing with microcontrollers. It is of moderate size offered at moderate to high cost. It has moderate power draw and response time, yet provides high accuracy, typically achieving precision levels of 0.1°C or better. Connectivity is achieved through a bridge circuit, allowing for resistance change using microcontrollers. Additionally, RTDs are favored for their stability and repeatability over long periods, which is essential for critical processes where consistent temperature monitoring is required.

Thermistors

Thermistors use the resistance-temperature relationship in semiconductor materials. It is small to moderate size and it is low cost. Its power draw is also low to moderate, with a response time ranging from moderate to fast. Accuracy varies depending on the type but can reach moderate to high levels, and it typically requires a simple resistor divider network for connectivity,

facilitating easy interfacing with microcontrollers. The flexibility in their design and cost-efficiency makes thermistors particularly suitable for consumer electronics and household appliances, where space and budget constraints are significant considerations.

Infrared (IR) Sensors

IR sensors measure the infrared radiation emitted by an object to determine its temperature. This sensor is moderately expensive and is larger than the previous ones. It offers fast response times and moderate accuracy, depending on calibration. Connectivity is good, with output available in analog or digital format, suitable for interfacing with microcontrollers. Additionally, IR sensors provide the advantage of non-contact temperature measurement, making them ideal for applications where the target object should not be disturbed or is inaccessible.

Semiconductor-Based Sensors

These sensors rely on the temperature-dependent properties of semiconductor materials. A very small temperature sensor typically offers moderate costs and low to moderate power draw. It provides fast response times and moderate to high accuracy, contingent on calibration. Connectivity is excellent, often featuring integrated circuits with digital outputs for seamless interfacing with microcontrollers. Their compact size and precision make these sensors particularly well-suited for use in medical devices and wearable technology, where space is at a premium and accurate temperature monitoring is crucial.

Gas Thermometers

Gas thermometers utilize the thermal expansion of gas to measure temperature precisely. It is large and expensive. Its power draw is moderate, and it exhibits slow response times. Despite this, it offers very high accuracy, achieving precision within millikelvins in controlled environments, and connectivity can range from fair to good, potentially requiring specialized interfaces for integration with microcontrollers. The exceptional precision of gas thermometers makes them invaluable in scientific research and calibration applications, where even minute temperature variations need to be measured with high reliability.

Bimetallic Temperature Sensors

Bimetallic sensors utilize the differential expansion of two metals to measure temperature. It is moderately sized and low cost. Its power draw is nonexistent as it operates passively without requiring additional power. However, it exhibits slow response times and moderate accuracy, typically within ±2°C, and with fair connectivity, allowing analog outputs to interface with microcontrollers. The simplicity and durability of bimetallic sensors make them particularly effective for use in household appliances such as ovens and thermostats, where long-term reliability is required with minimal maintenance.

Fiber Optic Temperature Sensors

Utilizing optical fibers, these sensors offer high precision and immunity to electromagnetic interference. Ideal for demanding environments, they ensure accurate temperature measurements.

They can vary from small to moderate size and can vary in cost, typically being high due to the precision and type. Its power draw is moderate, while response time varies from fast to moderate. Accuracy is generally high, with precision within a fraction of a degree Celsius, depending on the specific sensor type, and connectivity ranges from fair to good, often requiring specialized interfaces due to their optical nature when interfacing with microcontrollers. Optical fiber sensors are especially suited for applications in the aerospace and medical fields, where their resistance to harsh conditions and ability to function in areas with high electromagnetic fields or radiation levels are critical advantages.

Temperature Sensor Technology Comparison Table

	Size	Cost	Power Draw	Response Time	Accuracy	Connectivity
Thermocouple	3mm	$5-20	10mW	100ms	1C	Good
RTD	1-5mm	$30-200	20mW	2s	.1C	Good
Thermistor	5mm	$2-15	15mW	500ms	Varied	Good
IR	20mm	$50-500	50mW-1W	100ms	1-2C	Good
Semiconductor	3mm	$5-30	5mW-20mW	200ms	.5-2C	Excellent
Gas	1m	$200-1k	20mW	5s-2m	.0001C	Fair to Good
Bimetallic	10mm	$1-20	Passive	5s	2C	Fair
Fiber Optic	varies	$100-1k	1-10mW	2ms-2s	.01C	Fair to good

Temperature Sensor Technology Decision

Considering our group's parameters, the one device that fits our criteria is the Semiconductor devices. The RTD costs too much and its response time is too slow. Thermistor was a close second and might end being used if there are issues with the thermocouple. IR sensing was prohibitively expensive and was larger than we wanted. Thermocouple was another close option; We prioritize the response time over the other parameters but their connectivity leaves much to be desired. Gas is almost none of the above: large, expensive, and slow. Bimetallic was slightly too slow and inaccurate. Fiber optic was the best in most terms but is prohibitively expensive.

3.1.2 Temperature Sensor Product Comparison

Texas Instruments TMP1075

The Texas Instruments TMP1075 is a digital temperature sensor that incorporates a precision silicon-based sensing element and an analog-to-digital converter (ADC) on a single chip,

delivering digital temperature readings that are linearly proportional to the sensed temperature. The silicon temperature sensor ensures high accuracy and sensitivity across a broad measurement range. The device is designed to operate over a supply voltage range from 1.4V to 3.6V, demonstrating low power consumption with a typical quiescent current of just 10 μA, making it highly suitable for battery-powered and energy-efficient applications.

MCP9700T-E/TT

The Microchip MCP9700T-E/TT is a low-power, linear active thermistor IC that combines a temperature-sensitive analog output with a precision analog-to-digital conversion circuit on a single chip. This design simplifies its integration into systems by providing a voltage output that is directly proportional to the temperature, facilitating easy interfacing with analog input pins of microcontrollers. It operates across a supply voltage range from 2.3V to 5.5V, which, coupled with a low operating current of just 6 μA, makes it exceptionally well-suited for portable and low-power applications. Its tiny package size enables compact and flexible integration into a variety of designs, from consumer electronics to industrial controls. Offered by Microchip, the MCP9700T-E/TT is known for its cost-effectiveness and reliability in providing consistent and accurate temperature measurements, making it a popular choice for temperature monitoring solutions across a diverse range of applications.

Texas Instruments LMT87QDCKRQ1

The Texas Instruments LMT87QDCKRQ1 is a precision analog temperature sensor that combines a highly accurate, temperature-sensitive diode with a linearization circuit on a single chip, producing an analog voltage output that is inversely proportional to temperature. This configuration simplifies integration with analog-to-digital converters or direct monitoring through an analog input. Designed to operate over a wide supply voltage range from 2.7V to 5.5V, the LMT87QDCKRQ1 stands out for its extremely low power consumption, drawing a typical quiescent current of only 5.4 μA, which is ideal for energy-sensitive applications such as automotive and portable electronics. The sensor's compact SC70 package is optimized for efficient space utilization in designs where board real estate is at a premium. This sensor, known for its robustness and precision, is well-suited for temperature monitoring and control systems in a broad array of automotive and industrial applications.

Temperature Sensor Product Comparison Table

	Price	Accuracy	Response time
TMP1075	$0.24442	±0.25°C	~100ms
MCP9700T-E/TT	$0.30	±2°C	~150ms
LMT87QDCKRQ1	$0.346	±0.4°C	~150ms

Temperature Sensor Technology Decision

Considering our group's parameters, the one device that fits our criteria is the TMP1075 for its high accuracy and low power draw while also being easy to implement. This decision was reinforced by the TMP1075's compact size and I2C interface, which simplifies integration into existing systems and reduces overall design complexity. Furthermore, its compatibility with a wide range of temperatures makes it an ideal choice for our applications, ensuring reliable performance under varying environmental conditions. These features collectively make the TMP1075 a standout option, aligning perfectly with our project's needs for precision and efficiency.

3.2 Light Sensor

Light sensors are critical for the measurement of light intensity coming in through the user's window. We want to choose light sensing technology and products that will allow output a highly accurate numerical representation of visible light intensity that closely reflects the user's perception of light. With such data, we will be able to make accurate control decisions for the behavior of KnightTint.

3.2.1 Light Sensor Technology Comparison

Light sensing technologies emerged with the rise of semiconductor manufacturing. Using the existing diode, we managed to leverage the mechanics of silicon to accurately measure light. These light sensors produce a small voltage when light is measured and can be amplified and regulated to function with a host of controllers. In our project, we aim to use a light sensor on the outside of the window so that we can get accurate readings of the incoming light and be able to cool the room effectively. The main parameters set by our project are size, cost, current draw, response time, accuracy, and connectivity. Our devices need to be small enough to fit on the outside of the window without obscuring the view, cost less than $20, have a low current draw to efficiently run off the MCU, have a response time below 1 second, be accurate within 5%, and interface with our MCU well.

Photodiode

A semiconductor device that generates a current when exposed to light, where the amount of current is proportional to the intensity of the light incident on it, utilizing the photovoltaic effect. They are extremely cheap and small. They are highly accurate and consume very little power. Their response time is extremely fast and they require some external circuitry to convert their analog signal to a digital output.

Phototransistor

Similar to a regular transistor but with its base region exposed to light, causing an increase in conductivity and resulting in a collector current proportional to the incident light intensity, functioning as a light-controlled current amplifier. They are also extremely cheap but slightly

larger than photodiodes. They offer identical accuracy and power draw but slower response times. They have the same connectivity, requiring an ADC.

Light-Dependent Resistor (LDR)

A passive component whose resistance decreases with increasing light intensity due to the varying conductivity of the semiconductor material within it, making it act as a variable resistor based on light exposure. Following the previous devices, they are extremely cheap but larger than the previous. Their accuracy is lower but their power draw is the same. Their response time remains slower than photodiodes. Their connectivity is the same as the previous.

CMOS Image Sensor

Composed of an array of photodiodes integrated with CMOS transistors, where each photodiode converts incident light into electrical charge, which is then read out and processed to form an image, enabling digital imaging in devices such as cameras and image sensors. Their cost is much more expensive than the previous options and is typically larger. Their accuracy is much higher though, and their power draw is orders of magnitude larger. Their response time is slow but they have direct integration with MCUs.

Photonic Integrated Circuits (PICs)

Utilizes photonic components such as waveguides, modulators, and detectors integrated on a chip to manipulate and detect light, enabling functions such as wavelength filtering, modulation, and sensing with high precision and efficiency. They are by and far the most expensive option available. They are still in the same form factor while having the best Accuracy. Their power draw is also very dependent on the model but can be directly integrated with the MCU.

Light Sensor Technology Comparison Table

	Size	Cost	Current Draw	Response Time	Accuracy	Connectivity
Photodiode	1mm	$5-20	1 µA - 100 µA	100ns-10µs	1% - 5%	Good
Photo-transistor	2mm	$30-200	1 µA - 100 µA	10µs	1% - 5%	Good
LDR	5mm	$2-15	1 µA - 10 µA	10ms	5% - 10%	Good
CMOS	2-10mm	$50-500	1 mA - 10 mA	100µs-1ms	<1%	Excellent
PICs	1m	$50-500	1 µA - 10 mA	100ns-10µs	<1%	Excellent

Light Sensor Technology Decision

Considering our group's parameters, the one device that fits our criteria is the photodiode. The phototransistor offers some advantages but is more expensive and thus rendered obsolete. The LDR does not meet our design parameters for accuracy. The CMOS is too expensive and current hungry while being too large. The PICs are far too large to comfortably fit on a window.

3.2.2 Light Sensor Product Comparison

Texas Instruments OPT101

The TI OPT101 is a light sensor integrated circuit based on a straightforward monolithic photodiode and transimpedance amplifier (TIA) design. It integrates both the photodiode and the TIA on a single chip, providing an analog voltage output linearly proportional to the incident light intensity. The photodiode has a very large measurement area that collects a significant amount of light, allowing for high-sensitivity measurements. The IC accepts a power supply voltage between 2.7 and 36V, a low quiescent current of 120 μA, and small dimensions of 2.29 x 2.29 mm. This package is $6.25 sold by texas instruments.

Texas Instruments OPT4001 (SOT-5x3 Package)

The TI OPT4001 is a high speed, high precision digital ambient light sensor (ALS). It is part of TI's most recent line of ALS ICs and thus has more advanced functionality, better precision, and faster conversion times. The OPT4001 features a photodiode array configuration, with multiple photodiodes arranged in a grid pattern on the sensor chip. Each photodiode in the array responds to incident light, enabling the sensor to detect light intensity and variations across different regions. TI emphasizes that the spectral response of this sensor is very similar to that of the human eye, which they state is vital in use for human lighting experiences. Additionally, the IR rejection of this IC is exceptional, meaning the visible light intensity output is not confounded by infrared light. This sensor has a low operating current of 30 μA with an ultra-low power standby of 2 μA. The IC accepts a power supply voltage between 1.6 and 3.6V and has dimensions of 2.1 x 1.9 x 0.6 mm.

This sensor directly supports I2C communication with an on-chip analog to digital converter. For the SOT-5x3 package, the logarithmic 9 digit output encompasses a range from 437.5 μlux to 117 klux, which is broad and high enough to encompass direct sunlight intensity. This sensor also has built-in automatic full-scale range selection, so the microcontroller programmer does not have to select appropriate gain settings based on light levels. This package also has an additional interrupt (INT) pin that can raise an alert when a threshold is crossed, which can also be useful for the logic programmer. This sensor is $2.12 sold by Texas Instruments.

STMicroelectronics VL6180X

The STMicroelectronics VL6180X is a proximity and digital ambient light sensor. Like the OPT4001, this sensor is based on a photodiode array implementation. The sensor has an operating current of 300 μA for ALS and a standby current of <1 μA. The IC accepts a power supply voltage between 2.6 and 3.0V and has dimensions of 4.8 x 2.8 x 1.0 mm. The rangefining

technology in this sensor is unique and advanced, but that is not relevant for our use case. This sensor supports I2C communication with a wide dynamic output range between <1 lux to 100 klux. This is a high enough range for the use of measuring direct sunlight. This package also has additional pins for interrupts and measurement ready, which will be useful for microcontroller integration and programming. This sensor is $2.99 sold by STMicroelectronics.

Light Sensor Product Comparison Table

	Texas Instruments OPT101	Texas Instruments OPT4001 (SOT-5x3 Package)	STMicroelectronics VL6180X
Designation	Monolithic Photodiode	Digital Ambient Light Sensor	Digital Ambient Light Sensor
Maximum Lux	Design-Based	117 klux	100 klux
Output Protocol	Analog	I2C	I2C
Supply Voltage	2.7-36V	1.6-3.6V	2.6 and 3.0V
Cost	$6.25	$2.12	$2.99

Light Sensor Product Decision

We have decided to move forward with the TI OPT4001 for this project. For the lowest price of $2.12, it offers the most advanced functionality and high range. Its precision and response time are suitable for this project and its spectral response and IR rejection make it highly appropriate for a human-based system like ours. It accepts an appropriate voltage for our voltage supply configuration and has an extremely low operation current. We are choosing this over the OPT101 because we do not need the analog measurement of the light and would need to convert this to a digital signal anyway.

The OPT lacks its own analog to digital converter like the other two options have. We are choosing the OPT4001 over the VL6180X despite their similarities because the latter has many advanced rangefinding and IR emitter features that we simply do not need on this project. These features increase the hardware size and drive up the cost. Overall the OPT4001 will provide us with reliability, speed, and precision while facilitating integration with our system's logic.

3.3 Transceivers

In order to interface with the dimmer we intend to construct, we must interface with it using cables. Luckily for us, there are many different cables for many different forms of communication. We'll select the appropriate cable based on factors such as distance, data rate,

and environmental conditions to ensure reliable communication with the dimmer. Whether it's a twisted pair cable for Ethernet communication, a coaxial cable for analog signals, or a fiber optic cable for high-speed data transmission, our choice of cable will play a crucial role in establishing a robust connection with the dimmer.

3.3.1 Transceiver Cable Comparison

Coaxial Cable Transceivers

The Coaxial Cable transceivers are typically used in RF operations that require data transmission to be converted into radio signals. Coaxial cables consist of a central conductor surrounded by an insulating layer which is further encased by a conductive shield and then an outer insulating sheath. This stacked design allows for excellent shielding of the signal, making coaxial cables less susceptible to electromagnetic interference and RF interference. Only the central conductor sends the signal while all the other layers are used as shielding and grounding.

These cables are widely used in cable television networks, broadband internet access, CCTV systems, and high-frequency data transmission. One key advantage to these cables is their ability to carry high bandwidth and data rate signals. This makes them ideal for applications that require high-speed data transmission. They are also easy to install and connect, as they only need a BNC male and female connection.

Twisted Pair Cable Transceivers

These cables are a common type of transmission cables used in telecommunications and networking for transmitting analog and digital signals. They are constructed using insulated copper wire twisted together in a helical fashion. Each pair of wires carries a signal, one wire carrying the original signal and the other carrying the return path or acting as ground. The twisting of the wire pairs serves several purposes. Firstly, it helps to reduce electromagnetic interference (EMI) and crosstalk, which occurs when signals from one pair of wires interfere with signals on adjacent pairs. The twisting action ensures that any induced noise or interference affects both wires equally, minimizing its impact on the transmitted signal. Additionally, the twisting helps to improve signal integrity and reduce attenuation, allowing for longer transmission distances without significant signal degradation.

Twisted pair cables come in two main categories: unshielded twisted pair (UTP) and shielded twisted pair (STP). UTP cables are the most common and economical type, consisting of twisted pairs of wires without any additional shielding. They are widely used in applications such as Ethernet networking, telephone lines, and residential wiring. STP cables, on the other hand, have an additional layer of shielding, typically made of foil or braided metal, which provides enhanced protection against EMI and RFI. STP cables are commonly used in environments with high levels of electromagnetic interference, such as industrial settings and outdoor installations.

Twisted pair cables are categorized based on their performance characteristics, with different categories (e.g., Cat5e, Cat6, Cat6a) specifying parameters such as bandwidth, maximum data

rate, and maximum transmission distance. Higher category cables typically offer better performance and support higher data rates, making them suitable for demanding applications such as high-speed Ethernet networking and multimedia streaming.

Fiber Optic Cable Transceivers

Fiber optic cable transceivers are critical devices in modern communication networks, enabling the transmission of data over optical fiber. These transceivers convert electrical signals into a series of light pulses which are then transmitted through fiber optic cables. The core components of a fiber optic transceiver include a laser diode or a light-emitting diode (LED) and a photodiode. The laser or LED serves as the light source, converting electrical signals into light, allowing for high-speed data transmission over considerable distances without loss of integrity. This is pivotal in applications ranging from internet backbones to industrial networks where high data throughput and reliability are essential.

The versatility of fiber optic transceivers is further enhanced by their compatibility with various wavelengths and multiple protocols such as Ethernet, Fibre Channel, and InfiniBand. This flexibility ensures that they can be integrated into diverse network architectures, supporting a range of data rates and network standards. Additionally, the inherent advantages of optical fiber such as its immunity to electromagnetic interference and its higher bandwidth capabilities compared to copper cables make fiber optic transceivers a preferred choice for ensuring robust and efficient communication infrastructures.

Serial Cable Transceivers

These devices facilitate the conversion of data signals between serial communication forms, typically handling RS-232, RS-422, or RS-485 protocols. A typical serial cable transceiver consists of a driver and a receiver, which work together to transmit and receive data sequentially, or "serially", across a communication channel. This method of data transmission is particularly valued in industrial environments and in older computer networks where simplicity and cost-effectiveness are paramount. The robustness of serial communication protocols, coupled with the straightforward design of serial transceivers, allows for reliable data transfer even over longer distances and in electrically noisy environments.

The adaptability of serial cable transceivers to various industrial and commercial settings is enhanced by their ability to support multiple communication standards and interfaces. For instance, RS-485 transceivers are highly regarded for their capability to handle multipoint systems over long distances, making them ideal for networked industrial controls or building automation systems. These transceivers can manage data transmission across distances of up to 4000 feet by utilizing differential signaling, which significantly reduces susceptibility to noise and improves signal integrity. This feature, along with their ability to operate in half-duplex or full-duplex modes, makes serial cable transceivers a versatile solution for many low-speed, high-reliability data transmission needs.

Transceiver cable table comparison

	Ease of use	Cost	Reliability
Coaxial	High	Moderate	High
Twisted Pair	High	Low	Moderate
Fiber Optic	Low	High	Very High
Serial	Moderate	Low	Moderate

Transceiver Cable Decision

Based on the factors given, we have chosen to use Twisted Pair cables. They are incredibly easy to install onto a PCB, they have great reliability and low cost. This is due to their cheap material cost and ease of manufacturing. Several other options were weighed. Serial seemed to be a strong contender until their reliability appeared as their greatest downfall. Their degradation is faster than the other cables, leading to severe issues with receiving data. While fiber optic had the greatest reliability and speed, their high costs and low ease of use would cause major issues in our project. Their interfaces are incredibly difficult to interface with and would require much more expensive transceivers. Coaxial cables were close to our choice but their slightly higher cost acts as a deterrent for us. Given these considerations, Twisted Pair cables emerged as the most practical choice for our project's specific needs. Their flexibility and the relative ease with which they can be manipulated and installed on a printed circuit board (PCB) make them particularly appealing. Additionally, the simplicity of their connectors facilitates quick and error-free connections, further streamlining the development process.

3.3.2 Transceiver Product Comparison

The dimmable controller we intend to use uses an RS-485 protocol using a standard DMX cable. For our microcontroller to interface with the dimmable controller, we will need to implement

STMicroelectronics ST4E1240DT

The ST4E1240 is an advanced RS-485 transceiver designed by STMicroelectronics, aimed at industrial communication applications. It features high-speed data transmission, robust ESD protection, and a low-power consumption profile, making it ideal for harsh industrial environments. Additionally, the ST4E1240 supports a wide operating temperature range from -40°C to 85°C, accommodating extreme variations in industrial settings. It also includes fail-safe features that ensure reliable data communication even under conditions of electrical noise and signal interference.

Analog Devices Inc. ADM2761EBRWZ-RL7

The ADM2761E, by Analog Devices, is a robust, isolated RS-485 transceiver with integrated isolation. It's designed for reliable operation in electrically noisy environments, providing both data and power isolation for enhanced safety and signal integrity. Furthermore, the ADM2761E boasts a high data rate capability of up to 25 Mbps, making it suitable for high-speed communication needs in industrial applications. It also features a wide supply voltage range from 3V to 5.5V, which allows for flexible integration into varied system designs.

Texas Instruments SN65HVD3088ENSR

The SN65HVD3088E from Texas Instruments is an energy-efficient RS-485 transceiver, known for its low power consumption and extended battery life capabilities. It's suited for applications requiring long-distance communication with minimal power usage. Additionally, the SN65HVD3088E supports a wide input voltage range, ensuring compatibility with various industrial standards and improving its versatility in different environmental conditions. It also features robust electrostatic discharge (ESD) protection, safeguarding the device against common industrial electrical disturbances.

Transceiver Product Comparison Table

	No. of Pins	Cost	Voltage
ST4E1240DT	8	$2.14	3V to 5.5V
ADM2761EBRWZ-RL7	16	$2.60	5.7V
SN65HVD3088ENSR	8	$0.65	3V to 5.5V

Transceiver Decision

For our product, we compared the pin layouts of each and their prices. STMicroelectronics ST4E1240DT won out for its simplicity and its price. At $2.14 and being an 8-pin package, it will be easier to integrate into our PCB. We also considered the robustness of the thermal management features of the ST4E1240DT, which are superior for preventing overheating in tightly-packed electronic configurations. Its compact size and efficient layout support streamlined soldering processes, reducing overall labor for our devices.

3.4 Batteries

3.4.1 Battery Technology Comparison
AA Batteries

Alkaline

Alkaline AA batteries are one of the most common power sources in consumer electronics. They use zinc and manganese dioxide, as well as an alkaline electrolyte, usually potassium hydroxide, from which they derive their name. They output a steady nominal voltage of 1.5 volts, and have a capacity ranging from approximately 1000 to 3000 milliampere-hours (mAh). Alkaline batteries have widespread availability from major battery manufacturers and reasonable cost. Alkaline AA batteries are favored for their reliability and long shelf life, which can extend up to 5 years under optimal storage conditions. Their ability to perform under a wide range of temperatures and their leakage-resistant design make them suitable for a variety of everyday applications, from remote controls to portable radios.

Lithium

Lithium AA batteries utilize lithium-iron disulfide (LiFeS2) chemistry to output a steady nominal voltage of 1.5 volts. They have a battery life ranging between approximately 1500 and 3500 mAh, though when comparing the same brand, lithium batteries typically have higher energy density and battery life than alkaline batteries. Another very important benefit of lithium batteries is their relatively flat voltage dropoff curve during their life. They maintain a voltage close to their nominal voltage until they are nearly fully depleted. For these reasons, lithium batteries are generally more expensive. Like alkaline batteries, lithium batteries are also widely available from major battery manufacturers

Zinc-Carbon

Zinc-carbon AA batteries are an older, cheaper technology and are now less common than the previous two AA battery technologies. They output 1.5 volts. Zinc-carbon AA batteries have a lower battery capacity compared to alkaline or lithium batteries, usually ranging between 500 and 1500 mAh. Due to their cost-effectiveness, zinc-carbon AA batteries are often used in low-drain devices such as wall clocks and flashlights where high performance is not crucial. Moreover, while they do not offer the longevity or capacity of more modern battery technologies, their simplicity and accessibility continue to make them a viable option for budget-conscious consumers and in regions with less access to advanced battery types.

Nickel-Metal Hydride (NiMH)

NiMH technology is the standard for rechargeable AA batteries, which is their most appealing feature. They use a nickel oxide hydroxide cathode, a metal hydride anode, and an alkaline electrolyte. Their output is 1.2V, which is slightly lower than the standard 1.5V that the other battery technologies offer. NiMH batteries offer a slightly longer battery life than the standard

alkaline AA battery, ranging from 2000-3000 mAh, generally offering a battery life similar to Alkaline batteries at moderate-drain devices. NiMH batteries are the most expensive AA battery technology, but can be recharged hundreds of times, saving money in the long run.

Lithium-Ion

Lithium-ion batteries are a common technology used in rechargeable battery packs. This technology is not typically implemented in the AA form factor, though they do exist with limited commercial availability. Li-ion batteries use liquid lithium compounds as their electrolyte material, typically lithium cobalt oxide ($LiCoO_2$), lithium iron phosphate ($LiFePO_4$), lithium manganese oxide ($LiMn_2O_4$), or lithium nickel cobalt manganese oxide ($LiNiCoMnO_2$). $LiCoO_2$ is the standard, though there has been a recent shift towards more environmentally friendly alternatives. Lithium-ion batteries have a very high energy density compared to other technologies – 150 to 250 Wh/kg for li-ion batteries versus 100 to 150 Wh/kg for a standard alkaline battery. The most standard package for li-ion batteries is the 18650 size, which outputs 3.7V per cell. Li-ion batteries are generally more expensive than any of the AA technologies, and their implementation can be more complex.

Lithium-Polymer

Lithium polymer (LiPo) batteries are a type of rechargeable battery that uses a solid or gel-like polymer electrolyte instead of a liquid electrolyte found in traditional lithium-ion batteries. This allows for a more flexible and lightweight design. Like Li-Ion batteries, Li-Po batteries also most commonly utilize lithium cobalt oxide. Li-Po batteries have slightly higher energy density than Li-Ion batteries and also output 3.7V per cell. These batteries are often slightly more expensive than traditional Li-Ion batteries. Li-Po batteries do not have a standard package like Li-Ion batteries do, and are instead used when the shape has to be flexible.

Battery Technology Comparison Table

	Alkaline AA	Lithium AA	Zinc-Carbon AA	NiMH AA	Lithium-Ion 18650	Lithium-Polymer
Battery Life	1000-3000 mAh	1500-3500 mAh	500-1500 mAh	1300-3000 mAh	1500-3600 mAh	Size-Dependent
Rechargeable	No	No	No	Yes	Yes	Yes
Cost per Battery	$0.80-2	$2-4	$0.50-1	$1.50-3.50	$5-10	3500 mAh Example: $13-14
Nominal Voltage	1.5V	1.5V	1.5V	1.3V	3.7V	3.7V
Min. Series	3	3	3	3	1	1

Batteries Needed for MCU Input						
Dimensions L = Length D = Diameter	L: 50.5 mm D: 14.5 mm	L: 50.5 mm D: 14.5 mm	L: 50.5 mm D: 14.5 mm	L: 50.5 mm D: 14.5 mm	L: 65.2 mm D: 18.6 mm	3500 mAh Example: 129L x 56W x 4H mm

Battery Technology Decision

We decided to implement the 18650 Lithium Ion batteries in our KnightTint system. To decide on a battery technology for this system, we had to put ourselves in the shoes of the user. When one of the window batteries dies, we want them to be able to instantly switch it out for a new one with a full charge. In a non-removable rechargeable battery system, the user would have to plug in a charger to the battery based on implementation, which would add unnecessary complexity to the window boards. In a real-world implementation of KnightTint, there would be window PCBs located all around the home, and moving a charger to each window would be inconvenient. If multiple PCBs were to die simultaneously, it would also be inconvenient unless the user has multiple chargers. This reasoning eliminated lithium polymer batteries and most lithium ion batteries, We liked the replaceable nature of the AA and 18650 li-ion batteries

AA batteries are the most ubiquitous package in consumer electronics and offer the advantage of consumer familiarity and widespread availability. Li-Ion batteries, on the other hand, are more energy dense, which can help reduce PCB size.

Comparing the battery life offered by AA and 18650 batteries, we can see that the difference in battery life between the two technologies is not very large. With either route, we should be able to provide one month of battery life.

The nominal voltage output by each of these size standards is perhaps the most important aspect, as it makes a big difference in PCB size requirements. Most AA batteries output 1.3-1.5V, so with three of these in series, we can supply a nominal voltage of 3.9-4.5V, which we can regulate down to 3.3V. 3V is the minimum voltage for our system, but the AA's battery voltage drops below 1.5 V later in its lifetime, necessitating three. 18650 batteries, on the other hand, are 3.7 V, which we can regulate down to 3.3V without placing multiple in series. 18650 batteries are larger than AA batteries, but one 18650 battery is smaller than three AA.

18650 batteries are more expensive than AA batteries upfront, but the need for less of them and their rechargeable nature makes up for this higher cost. Overall, 18650 lithium-ion batteries are the most appropriate choice given our goals of user convenience and simplicity for the KnightTint system.

3.4.2 Battery Product Comparison

Samsung SDI 35E

The Samsung SDI INR18650-35E is the highest capacity 18650 offered by Samsung SDI, coming in at 3500 mAh. Samsung SDI is a subsidiary of Samsungs, which is a well known company that is reputable in terms of their quality and honesty.

The nominal voltage for these batteries is 3.6V, which is slightly lower than other 18650 models, but is still acceptable for our 3.3V application. The maximum voltage at full charge is 4.2V and the cutoff voltage is 2.5V. The battery voltage will drop off towards 2.5V at the end of its discharge cycle, so we will not get the full 3500 mAh battery life at 3.3V, which is why we are examining the highest capacity model. The battery is rated for 8A continuous discharge, which is well above our current needs. Its estimated charging time, which we want to be as short as possible, is 4 hours, charging at 2000 mA. The price per battery at low purchase quantities is $4.65, which is slightly below the expected cost per 18650 battery.

SKYWOLFEYE 18650 9900 mAh

The SKYWOLFEYE 9900 mAh is one of the the top results on Amazon for 18650 batteries and was extremely intriguing due to its obscenely high advertised battery life rating. Users of KnightTint that seek to replace their batteries may see this product and think that it is a good option. However, after further research, we have concluded that this product is a scam and does not provide nearly 9900 mAh of battery life. In the words of an internet forum user, these batteries are "not worth the plastic they are printed on". We recommend that users avoid these batteries, and we will not be selecting it for the development of our prototype.

Vapcell N40

Vapcell is the best selling brand from vendors dedicated to selling 18650 batteries, which tend to carry more reputable brands. The consensus about Vapcell is that they make a good quality battery, but tend to slightly exaggerate their mAh ratings.

The Vapcell N40 is an 18650 battery with an advertised battery life of 4000 mAh. The nominal voltage is 3.6V, ranging from 2.5 to 4.2V, much like the Samsung SDI 35E. This battery is rated for 10A continuous discharge, and charges at 2500 mA. The price per battery at low purchase quantities is $7.99.

Battery Product Comparison Table

	Vapcell N40	**SKYWOLFEYE 18650 9900 mAh**	**Samsung SDI 35E**
Reported mAh	4000	3500	9900
Brand Trustworthiness	Medium	Low	High
Nominal Voltage	3.6	3.7	3.6
Price Per Battery	$7.99	$4.00	$4.65

Battery Product Decision

We have chosen to proceed with the Samsung SDI 35E 18650 battery. This battery is from a trusted brand, has a high capacity, good voltage characteristics, and appropriate current ratings. While the Vapcell N40 has slightly better capacity, the 35E is about 60% of the cost for similar performance. We will need six batteries for our project (not including any backups), so the total savings will be significant.

3.5 AC/DC Power Supply

Our central MCU that gives directions to the window custom PCBs will need a DC power supply, and we do not want to use a battery to power this board. Thus, we need a AC/DC power supply. For this reason, we have decided to opt for a reliable and efficient AC/DC power supply that can handle a wide range of input voltages and provide stable output for the MCU. This will ensure that our central control unit operates continuously without the risk of power interruptions or fluctuations that could affect the performance and reliability of the window control system.

3.5.1 AC/DC Power Supply Technology Comparison

Switched-Mode Power Supply (SMPS)

Buck Converters

Buck Converters, also known as step-down converters, are a type of switched-mode power supply (SMPS) that decrease the input voltage to a lower output voltage. They operate by intermittently turning on and off a switch (typically a MOSFET) to control the energy flow to an inductor, which then transfers energy to the output. Buck converters are widely used in battery-powered devices, voltage regulation applications, and power supplies where a lower output voltage is required. They are efficient and relatively simple, making them suitable for a wide range of applications. They are commonly used in low to medium voltage applications, typically operating from a few volts (battery-powered devices like smartphones and portable

electronics) to several tens of volts (automotive systems and voltage regulation in power supplies).

Flyback Converters

Flyback converters are another type of switched-mode power supply (SMPS) that store energy in an inductor during the on-time of a switching transistor and transfer it to the output during the off-time. They contain a transformer, allowing them to provide step-up, step-down, or isolation voltage conversion depending on the winding configuration. They are commonly used in low to medium power applications such as consumer electronics, LED drivers, and offline power supplies. Flyback converters inherently provide electrical isolation between the input and output, making them suitable for applications where isolation is required. Due to the additional components required for the transformer and isolation, Flyback converters are generally more complex than buck converters.

Resonant Converters

Resonant Converters utilize resonant circuits, typically consisting of inductors, capacitors, and switches, to achieve high efficiency by operating at specific resonant frequencies. Resonant converters exploit the energy stored in the resonant components to reduce switching losses and improve overall efficiency. This is achieved by allowing the circuit elements to resonate with each other, resulting in zero-voltage or zero-current switching, which minimizes power dissipation. A common type of resonant converter in an LLC resonant converter, in which the resonant elements are two inductors and a capacitor. Resonant converters are commonly used in high-power applications where efficiency and power density are critical, such as telecommunications, renewable energy systems, and electric vehicle chargers. They can be larger and more complex compared to traditional SMPS, requiring careful tuning of the resonant circuit parameter. Resonant converters are not readily available as stand-alone 120VAC-3.3VDC modules and are instead available as reference designs or utilized in advanced settings and by custom power solutions companies.

Linear Power Supplies

Linear Power Supplies regulate voltage by dissipating excess power as heat using linear regulators. They are typically used in low to moderate voltage applications, ranging from a few volts to tens of volts. They are relatively less efficient, especially when the input voltage is significantly higher than the output voltage. They are more commonly used where low noise, simplicity, and low to moderate power levels are more critical than efficiency, such as audio equipment and instrumentation. Most linear power supplies are variable, typically allowing the user to vary the output voltage with a knob or switch. These power supplies are typically much larger than AC-DC converter modules and cannot be integrated into a PCB.

AC/DC Power Supply Technology Comparison Table

	Buck Converter	Flyback Converter	Linear Power Supply	Resonant Converter
Functionality	Step-Down	Step-Down Step-Up	Step-Down	Step-Down Step-Up
Complexity	Low	Medium	High	High
Commercially Available Pre-Fabricated	Yes	Yes	Yes	No
Cost	$5-15 (Low)	$6-20 (Low)	$50-150 (High)	Medium

AC/DC Power Supply Technology Decision

We have decided to move forward with a power supply based on buck converter technology. For our project, we just need a way to convert 120 VAC to 3.3 VDC, and all of the above technologies are capable of performing this operation. Immediately, we eliminated the linear power supply because it is far too large to fit on our circuit boards and is a magnitude more expensive than its alternatives. This then left us to decide between the three switched-mode power supplies.

We decided that the resonant converter was more complex than necessary for this project. The additional complexity would make the project more expensive and add more components that could potentially break. We would also have to use the reference design to create this power supply rather than purchase it. Between the buck and flyback converter, we decided to eliminate the flyback converter because we do not need the step-up functionality. The additional transformer would increase hardware dimensions and cost. The buck converter is the most popular technology for step-down AC/DC conversion for a good reason and is an appropriate simple solution for the use case in our project.

3.5.2 AC/DC Power Supply Technology Product Comparison

Mean Well RS-15-3.3

This power supply is a buck topology-based power supply from Mean Well, which is a very established power supply brand. It is capable of converting 85-264VAC to 3.3VDC and supports an output current of up to 3A. The documentation on this product is very rigorous, with technical and safety compliance information readily available. It adheres to all relevant safety and compliance standards. The dimensions of this power supply are larger than its alternatives at 62.5 x 51 x 28 mm. However, this is due to a fully enclosed design and the inclusion of the input

terminals at the bottom. The efficiency given by the manufacturer is 72%, and this product is $8-10 depending on the retailer.

NOYITO NOADPB33700

This product is a buck power supply module capable of converting 100-264VAC to 3.3VDC. It can support an output of up to 700 mA, which will be more than enough for the expected current draw of our controllers. This power supply has a compact and fully open design, with dimensions of 29.5×20.1×17.5 millimeters. The efficiency given by the manufacturer is 80%, and the cost of this power supply is $7.00. Additionally, the robust design of this power supply module includes over-current and short-circuit protection features, enhancing the reliability and safety of its operation in our controllers.

AmazonBasics USB-A Wall Charger

This is a more basic solution to our voltage converter needs. This is a simple USB charging block that is commonly used to charge smartphones, but is also applicable for our project. This charger plugs into the wall and outputs 5V from the USB-A port. We may need some additional hardware to interface with the USB cable and will need to find a development board that accepts 5V input voltage. There are hundreds of generic manufacturers of these chargers, but we are choosing AmazonBasics because we can count on Amazon to provide a quality product for a good price - $7.

AC/DC Converter Product Comparison Table

	Mean Well RS-15-3.3	**NOYITO NOADPB33700**	**AmazonBasics USB-A Wall Charger**
Output Voltage	3.3V	3.3V	5V
Efficiency	72%	80%	Not Provided
Hardware Integration Complexity	Medium	Medium	Low
Cost	$9	$7	$7

AC/DC Power Supply Product Decision

We have decided to move forward with the AmazonBasics wall charger as the AC/DC converter that will power our development board. This will keep the hardware simple, cost-effective, and familiar for the user.

3.6 Microcontroller

Microcontrollers can be put into various categories based on the different features and specifications that they offer. In the sections below, we will explore these categories and compare their features to determine which one is best suited for our tinting system project. Selecting the right microcontroller is crucial for the success of our project and to ensure its timely completion. Therefore, we need to conduct a thorough analysis of each category's unique features and capabilities.

3.6.1 Microcontroller Technology Comparison

8-bit, 16-bit, and 32-bit Microcontrollers

Microcontrollers are categorized based on the size of their data bus and registers. 8-bit microcontrollers are generally used for simpler applications that don't require much processing power. They have a limited addressing range and can only perform arithmetic operations that result in numbers ranging from 0 to 255. They are slower compared to their 16-bit and 32-bit counterparts, but they are budget-friendly and sufficient for many low-end applications such as remote controls, toys, and simple sensors.

Moving up to 16-bit microcontrollers, some of the advantages and disadvantages of 8-bit microcontrollers still hold. However, the 16-bit controller is faster and can handle larger numbers, making it more suitable for complex applications that require more processing power, such as motor control, data logging, and advanced sensors. They are also useful in applications that need to manage large datasets or handle complex algorithms. Although the 16-bit microcontroller can be more expensive than the 8-bit microcontroller, the price increase is still worth it.

Finally, 32-bit microcontrollers have a much larger data bus than 8-bit microcontrollers, making them more data efficient and faster in processing information. However, they are more power-hungry compared to 8-bit and 16-bit microcontrollers. While they are suitable for more demanding applications, they may not be necessary for simpler processes.

General-Purpose vs. Application-Specific Microcontrollers

Microcontrollers can be designed for specific applications, such as motor control, automotive, or medical devices, or they can be general-purpose and versatile for a wide range of applications. Each design approach has its own advantages and disadvantages. General-purpose microcontrollers are designed to support a wide range of applications, making them very flexible and versatile. They are usually accompanied by detailed documentation, which makes maintenance straightforward. They are also cost-effective because they are cheaper to purchase than a design-specific processor, which requires a lengthy design process. However, because they are so general and versatile, they may include unnecessary features for a specific project, leading to inefficiencies and higher power consumption.

The use and implementation of these systems in various applications can help us understand where general-purpose and application-specific microcontrollers are used. In consumer

electronics, such as toys, cameras, and robotic devices, general-purpose controllers are commonly used. For example, remote-controlled cars or automobile systems for engine control and sensor data use general-purpose controllers. Application-specific microcontrollers are used in medical devices such as electrocardiography (ECG) machines. These microcontrollers are designed to be accurate and efficient in processing the specific signals generated by the human heart. Aviation and military applications also use these systems for tasks such as navigation and controls.

Low-Power Microcontrollers

Low-power microcontrollers, as their name suggests, are a group of microcontrollers that focus on being more power-efficient. This can be achieved through various processes, one of which is a sleep mode where certain parts of the board are disabled, waiting for a specific event to raise an interrupt and wake up the microcontroller to perform the required task. This mode helps reduce the power consumption that often occurs when the microcontroller is idle and waiting for a specific event.

Low-power microcontrollers also come with the advantage of being more cost-effective, making them a great choice for tasks where the amount of energy available is limited, for example, when the device has to run on a battery. This is a crucial factor to consider when choosing which type of microcontroller to use for a system that relies on batteries as the primary power source. However, with all the efficiency considerations, they also come with some disadvantages, such as having limited resources like memory. To save on power consumption, reducing the amount of resources being kept track of can be very useful.

When we look at where most of these low-power microcontrollers are in use, we can get a broader idea of whether we want to narrow our scope to focus on efficiency for our system. These microcontrollers are commonly found in systems like fire smoke detectors or fire alarms, all of which need to be constantly on and consume as little power as possible. In most buildings, fire alarms are triggered to be battery-powered in the chance that there is no power so that people can still be warned and get to safety.

Digital Signal Controllers (DSCs)

Digital Signal Controllers (DSCs) are specialized microcontrollers designed to process digital signals efficiently. They combine the features of microcontrollers with the ability to do some digital signal processing in one chip. These chips offer several advantages, including fast interrupt responses and control peripherals such as PWMs and watchdog timers. They are typically programmed using the C language. While they can be more expensive than regular microcontrollers, they are less expensive than computer chips that can perform the same task. They are designed to perform real-time processing which makes them extremely useful for devices that require a fast response time.

However, there are some disadvantages to using DSCs. For example, they are easily damaged by static charges, so this needs to be taken into account when deciding how to use them. They also cannot directly interfere with higher-power devices, which is another factor to consider when using them in a design. Despite these limitations, DSCs are used in many different applications,

including motor control and sensor processing. The features they provide align with some of the things we are looking to do for this project, but we need to be aware of their limitations when using these chips.

Networked Microcontrollers

Networked Microcontrollers are a type of microcontrollers that are specifically designed for networked applications. They are equipped with communication interfaces such as Ethernet, Wi-Fi, or Bluetooth. These microcontrollers are designed similarly to regular computers but in a smaller format that can operate smart devices by providing the necessary power, memory, and input/output for the peripherals. Networked microcontrollers can provide sufficient computing power while keeping the price and energy usage low. This is especially important for smart devices that are powered by batteries and require power efficiency. Additionally, their small footprint makes them well-suited for smaller electronic devices.

Networked microcontrollers have a wide range of applications, as seen in the recent influx of smart home devices that allow for a more connected home. They are also used in other devices such as ATMs, home security systems, and card swipe machines, all of which require a network connection to provide their advertised services. This makes using them in a smart Windows system a sensible choice. It will make it easier to integrate network features such as syncing through Wi-Fi and other smart home connection features that we plan to implement.

Safety-Critical Microcontrollers

Safety-critical microcontrollers are a group of microcontrollers that are used in systems where failure could result in injury or loss of life. They are designed to maintain high-reliability standards and are used in applications where safety is the number one priority. To achieve these high safety standards, these chips use various architecture and techniques to prevent and detect the random errors that can occur with microcontrollers. While some of these features can be implemented in other microcontrollers, these controllers are specialized in making the process a lot simpler and easier to implement. They provide safety at a hardware level rather than through software.

As microcontroller devices become more complicated, it becomes harder to implement some of the hardware safety features mentioned before. With the increase in security, there is also a rise in cost, making them a lot more expensive to buy. These microcontrollers are often accompanied by their separate software to ensure security, which can also add to the overall cost of these controllers. While the cost of these chips is always a consideration, their application use is often necessary or even mandated in things like vehicle safety-critical systems like powertrain, power steering, and airbag systems. They are also used in electrically controlled systems like planes and helicopters where safety is the number one priority.

3.6.2 Embedded Software Technology Comparison

C

The C programming language was originally developed in the early 1970s by Dennis M at Bell Laboratories as a sample language for writing operating systems for small computers. Over the years, C gradually gained popularity and became one of the most widely used programming languages. Compilers became available for all types of computer architectures and operating systems. C is particularly popular for embedded software, where it is used widely in many embedded systems due to its efficiency. Microcontroller systems often rely on C for their efficiency as they are often limited by their hardware. C is also a great choice for moving from one platform to another as it requires very little change to achieve compatibility. C is a lower-level language that provides greater access to the hardware, making it very attractive for embedded systems where control over hardware is required. C is used widely throughout the industry, estimated at 80% of all embedded systems.

However, there are some disadvantages to using C in embedded systems. For instance, it lacks some features that come with higher-level languages such as exception handling and runtime checking. Memory management is another consideration as compared to higher-level languages, C requires manual memory management which can lead to errors that need to be handled properly. This can be an advantage and a disadvantage as memory can be a limitation in many of these systems and being able to handle it properly can determine the limitation to what you can achieve with these systems.

C++

The C++ programming language was developed in the 1980s to add object-oriented programming features to the C language. C++ is considered a middle-level programming language as it combines aspects of both higher and lower-level languages. It is directly related to C, with added features like class objects and exceptions. C++ maintains much of the same efficiency as the regular C language, but it also has libraries that can save users a lot of time when implementing them. Its relationship with C also means that it retains its portability and ability to move from one platform to another. The higher-level features added to C++ have increased its complexity and ease of use for users when implementing it in an embedded system.

Python

Python is a popular high-level programming language that was created in the 1980s. Its design prioritizes code reliability and emphasizes syntax that allows programmers to express concepts in fewer lines of code. Python has a vast range of libraries that provide excellent support for users and make it easy to implement higher-level concepts. Some of the libraries include micropython, which is a more efficient and less resource-intensive version of the Python 3 language, making it easier to run on microcontrollers with limited resources. Other resources like Zerynth make it easier for internet-integrated microcontrollers to connect to popular cloud platforms. However, because of its higher-level features, Python may face speed issues, and it may require a significant amount of memory, which can be a limitation for embedded systems. Python's unique set of features limits its usability for some users. While its ease of use can be attractive, the

overhead that it requires can be overwhelming for systems that prioritize simplicity and low power consumption.

Java

Java is a popular programming language that is widely used in embedded systems due to its portability, code reusability, reliability, and security features. Java offers excellent cross-platform compatibility, allowing software to run seamlessly on different platforms, including embedded systems. Over the years, Java has undergone significant improvements, making it a highly reliable language for use in embedded systems. Java is also known for its simplicity, making it a great choice for larger projects. However, it can be complex due to the numerous packages and classes it offers, which may overwhelm programmers. Compared to other languages like C, Java is slower since it is a high-level language and has less integration with hardware.

JavaScript

JavaScript was first introduced in 1995 as a programming language mainly for developing web pages. Although it is not traditionally used in embedded systems, it has started to be used in some projects for these systems. In such projects, JavaScript is used as an interpreter to provide information from the board and an API for hardware-specific functions. When used, JavaScript is mainly chosen for its HTML5 support, which is a good option for systems that require significant networking or graphics. However, there are certain factors to be considered when using JavaScript, such as its poor runtime efficiency, which can be a concern for embedded systems. The language also uses dynamic typing, which can introduce a significant memory overhead, making it unsuitable for some embedded systems. Therefore, JavaScript is useful for specific systems where it is necessary, and the overheads can be managed to avoid impacting the overall system performance.

Embedded Software Technology Comparison Table

Language	Characteristic	Pros	Cons
C	Procedural language that is known for efficiency and control over system resources.	- Powerful and efficient - Portable - Built-in functions - Base language for other languages	- Data Security - Lack of features like exception handling and runtime checking -No memory management
C++	An extension of the C language that adds higher-level features like object-oriented programming.	-As efficient as its predecessor. -A lot of resources available due to its popularity -Very powerful and can be used to create a lot of higher-level programs	-The higher-level feature comes with a higher level of complexity when it comes to implementing it in an embedded system.

Python	Interpreted higher-level language with simplicity.	-Simple and easy to learn. -Versatility and adaptability to all types of systems. -Code is easy to understand	-Difficult to develop for large applications - Large overhead for space and hardware - Speed issues - Usability limitations
JAVA	An efficient, multi-purpose language used extensively for internet-based applications and object-oriented software.	-can move from device to device with ease. -Object Object-oriented programming. -Multithreading	-Slow execution speed - Memory management -Does not go as low level as other programming languages
JavaScript	Higher-level scripting language mainly for developing web pages.	-A good solution if your system requires the use of HTML5	-Poor runtime efficiency -Not easy to maintain

3.6.3 Communication Protocol Technology Comparison

Communication protocols are a set of rules that allow two or more devices to exchange information. In the realm of microcontrollers (MCUs), these protocols are essential for facilitating communication between the MCU and other devices or components within your system. The choice of protocol depends on various factors, such as the need for one-way or two-way communication and the desired speed of data transmission. Some common communication protocols used in MCUs include I2C, UART, and SPI. In the following section, we delve into these different protocols and explain how they function.

I2C

The I2C communication protocol establishes a connection between two or more integrated circuits. It requires two wires to send and receive information, and it supports the serial transfer of data. Multiple devices can be connected using a master-slave configuration.One of the advantages of using I2C is that it can be configured to have multiple masters, which is especially useful when there are many devices connected. It also reduces complexity as only two wires are required for implementation. However, there are some drawbacks to consider. The speed of I2C may not be as high as with other protocols. It is set up for half-duplex communication, which is something to keep in mind. Additionally, the hardware required to set up multiple slaves increases in complexity with the number of slave and master devices that are set up

UART

The Universal Asynchronous Receiver Transmitter, also known as UART, is a hardware device that facilitates communication and information sharing with other devices. It sends and receives data bits sequentially, each framed by a start and a stop bit. UART can be configured with

different parameters, such as the baud rate, which determines the speed of data transfer, and flow control. When it comes to communication, UART is straightforward and does not require complex exchanges or a clock signal. It only requires two wires, making it easy to implement. However, these advantages come with certain limitations. For instance, the size of the data being transmitted and the speed of data transfer are less because the transmission has to occur sequentially

SPI

The Serial Peripheral Interface (SPI) is a communication protocol known for its high-speed applications. It is a full-duplex implementation, meaning that data can be sent and received on the same connection simultaneously. One of the most attractive aspects of SPI is its speed and versatility across different applications. It supports rates of up to 100 Mbps, which is more than sufficient for most applications and is not a necessity for the sensors and other applications we plan to use for this project.

SPI also supports a push-pull method that allows for longer communication distances. While there are many advantages to using SPI, there are also some considerations to keep in mind. For instance, SPI does not have flow control or acknowledgment features. Additionally, unlike with other communication protocols like I2C, the master-slave relationship cannot be altered in SPI.

MCU Interface Protocol Technology Comparison Table

	UART	I2C	SPI
Complexity	Very low	Low, even with many devices	Complexity rises as more devices are added
Minimum Wires Required	2	2	4
Duplex	Full-duplex (send and receive over the same connections at the same time)	Half-duplex (send and receive over the same connections but not at the same time)	Full-duplex (send and receive over the same connections at the same time)
Type Of Communication	Asynchronous (messages do not need to be consistent in length)	Synchronous (all transmissions look the same, a requirement for synchronous buses)	Synchronous (all transmissions look the same, a requirement for synchronous buses)
Number Of Targets	Connections are normally device-to-device	Up to 128 when using a 7-bit address, with a default of 27	In theory, a serial chain can support any number of targets

Number Of Controllers	Connections are normally device-to-device	Multiple controllers can be supported	1
Distance	Up to 1000 m	Up to 100 m	Over 10 m
Speed	5Mbs	5Mbs	100Mbs

3.6.4 Microcontroller Product Comparison

We are planning to implement an MCU in our system that can receive information from around four sensors and interpret that data using algorithms and logic. Based on the data, the MCU will take different actions, such as adjusting the voltage to increase or decrease the tinting value of the windows. We don't need a lot of processing power for our MCU, but we do need it to have WiFi implementation so that we can communicate between windows. We also need to have enough pins on our board to be able to read all the information from all the different sensors. These are our main requirements for selecting an MCU, and in the following sections, we will explore different options and choose the best one for our project.

MSP430

The MSP430 family of microcontrollers from Texas Instruments is very versatile and can be powered based on the choice made when it comes to the specific chip. They are usually 16-bit MCUs and are often very affordable solutions for a variety of different types of applications. The over 2000 different types of chips that are available in this family make it so that there are options for all the different types of applications. With all these different options available, there is still the problem of the chip not natively supporting Wi-Fi. For this project, Wi-Fi is a must-have, and we would have to implement the Wi-Fi communication using a separate module from the chip, which can raise the level of difficulty when it comes to the implementation within those systems. While the lack of Wi-Fi support can seem like a downfall for these chips when it comes to this specific use case, they do support the I2C protocol of communication, which is a must when looking at connecting different sensors and being able to implement them together. These chips can be programmed and controlled using Code Composer Studio, which is the specific environment provided by Texas Instruments, and within that suite, C is mostly the language that is used.

AVR

The AVR group of MCUs, just like the MSP family, comes with a variety of different chips, each having a variety of different features based on the specific project that you are looking to implement them in. For example, when looking into the AVR EB section of these chips, they have very powerful CPU cores that can process and provide very accurate information and be optimized to provide a lot of performance while at the same time being cost-effective. Wi-Fi compatibility is not something that comes with a lot of these chips. Any chip from this family will have to be checked for Wi-Fi compatibility to make sure that it will end up working within our project. The number of pins is something that can also be a concern when it comes to this

49

family of chips, so making sure that the right amount of pins is available for all of our sensors is a must. When it comes to actually programming these chips, most of the code is written in the C programming language due to some of its low-level features useful when it comes to a microcontroller.

PIC (Peripheral Interface Controller)

Developed by Microchip Technology, PIC microcontrollers are known for their low power consumption and are often used in industrial and consumer electronics. These are mostly 8-bit microcontrollers that enable a lot of low power consumption for uses like real-time controls. While these microcontrollers come with the benefits of low power consumption, this does come with the unavailability of features like Wi-Fi. If we were to use one of these chips, we would have to implement it using a separate module to be able to add the feature we are looking to add. Looking at it from the perspective of adding the Wi-Fi unit, it will take away the main advantage that comes with using these types of chips, which is the low power consumption. These chips, while very useful for tasks that require power consumption awareness and do not need much processing power, will not be useful when it comes to our project as they lack some of the features that we are looking to add.

ARM Cortex

ARM-based microcontrollers, such as the STM32 and NXP series, offer a wide range of performance levels. They are prevalent in industrial and automotive applications. These are known for their high performance, power efficiency, and various connectivity options available for different application uses. The availability of Wi-Fi depends on the specific MCU and whether they have an implementation for it because there are many different types of chips that come with all types of peripherals. When looking at this family, it would be a choice for the different needs that we have in the project. There are a lot of manufacturers that make the ARM Cortex type of MCU, with Texas Instruments being one of them. As we are already familiar with a lot of software and how to program the Texas Instruments chips, this would save a lot of time from having to learn a lot of this software from scratch.

ESP8266/ESP32

The ESP8266 and ESP32 group of microcontrollers are developed by Espressif and are known for their high performance, power efficiency, and the various connectivity options that are implemented within these MCUs. When looking at the implementation for this project specifically, there is not a lot of processing power that is needed to accomplish the task that would make this project successful. The ESP family of chips does provide the processing power should it be needed to implement the logical algorithms that are needed to make this project successful. When it comes to connectivity, these chips are a perfect fit with their built-in Wi-Fi modules, making them perfect for our wireless communications needs. These chips also come with a variety of different GPIO pins, allowing us to connect all the different sensors and input information that are needed for this project. These chips, with all their processing power and various communication abilities, will make for a great chip to use in the windows.

FPGA (Field-Programmable Gate Array)

One last group that we should take into consideration, while not a microcontroller, FPGAs provide a different approach to embedded systems by allowing hardware customization. They are essentially a large group of logic components that can be arranged to perform tasks with varying complexity. This makes them very capable and can have a great advantage when looking at specific tasks. While that processing power can be significant, implementations of things like Wi-Fi can be very complex on an FPGA and can require outside components to be able to achieve it. Beyond the hardware itself, the algorithms and programs used in FPGA use Hardware Description Languages like Verilog. These languages are very different from the higher-level languages and can have a higher learning curve. While some of our group members have had some coursework within Verilog, we do not have a great deal of expertise and will require us to learn the whole thing again. These chips can perform very well in certain applications but when looking at our projects and our needs, we will need to go with something else that would better suit those needs.

Microcontroller Family Comparison Table

Microcontroller Family	Wi-Fi Support	I2C Support	Programming Language	Power Consumption
MSP430	No (requires external module)	Yes	C	Moderate
AVR	Varies (check specific chip)	Yes	C	Low
PIC	No (requires external module)	Yes	C	Very Low
ARM Cortex	Varies (check specific chip)	Yes	C	Varies
ESP8266/ESP32	Built-in	Yes	C	Moderate
FPGA	Complex (requires external components)	Can be implemented	Hardware Description Languages (Verilog)	Varies

Main Microcontroller Product Decision

After looking into all the different MCUs that are available to us, their benefits, and shortcomings, we are now going to make a decision on which one will be best for our use case in this project. To do that, we will look at the processing power and how much the chip is going to have. We will then look at the different price ranges and compare the price-to-performance ratio, choosing the best performance for the best price. We will also look at the list of features that are

offered for that chip, choosing the one that offers the most features that could be useful for our project. While the language is not always a choice, a board that works with C, which a lot of our team has experience with, would work well to help with one of the constraints that we have, which is time.

When looking at the higher-level categories that we have, we know that when it comes to the processing power that we are going to need, we will need an MCU with at least 16 bits to make sure that we are able to handle the amount of processing that we will need. We will go with a general-purpose controller rather than anything specific because of some of the issues that come with having an application-specific MCU. The tasks that we will need the MCU to accomplish should be available on most general format MCUs in the market. Power consumption will not be taken into consideration when making this choice as AC power will be used to power most of this system. As described in the device specification, our windows will be Wi-Fi connected so our MCU will need to have some form of network connectivity feature. While this feature can be added later on to an MCU that doesn't have it, we would rather it be integrated as one system so that we would have fewer points of failure within our system.

When going into the specific type of MCU, one choice stands out the most as being the obvious choice when it comes to our use case for this project. The MSP430FR6989 is a perfect selection with more than enough pins and enough power to perform all the needed tasks for this project. The MSP430FR6989 does not natively support wifi but we can add it with the CC3100 booster pack. This will make it easy based on our previous experience with the MSP430FR6989 making sure that everything is integrated correctly.

Window Microcontroller Product Decision

After choosing our main board, it comes time to select our other MCU that is going to go on the window boards that will collect the information from the sensors and send it via wifi to our main hub system. Due to the simplicity of the task, we are going to be choosing one of the simplex MCUs so that we can keep the whole system as simple as possible. Since we will be the ones implementing the PCB for this MCU we do not have to have the wifi implementation within the MCU itself we can add a module on the board that would take care of that for us. When looking at the MSP430 class of controllers any of them could work for this task as long as they meet our feature needs like having I2C communication and being able to connect to a wifi module.

3.7 Full Stack Development

In this section, we explore diverse software options that can be used in the Smart Tinting House. Software plays a pivotal role in building our project to interact with modern technologies and provide tangible outcomes with real-world impact. It serves as the backbone for implementing theoretical concepts into practical solutions and enabling experimentation, analysis, and dissemination of findings. Each subsection of this main section contains the selection of the corresponding software, followed by the implementation of the software into the project, and finally examining the advantages and disadvantages of each choice that could make the design process more achievable.

3.7.1 Mobile Device Platform Comparison

In the new era of technology, mobile applications are no longer considered application tools for marketing products, they are necessities. With the increase in use of smartphones, mobile applications allow people to engage not only with others but also with other devices, making the integration of a mobile application into the Smart Tinting House project an important feature.

Building such an app for our design requires software program advancement and user interface abilities. The mobile device must run a user interface that permits users to control and monitor the tint of the window. For illustration, users ought to be able to alter and plan the level of tints based on the time of the day or any particular day of the week.

Apple (iOS)

iOS (formerly iPhone OS) is a mobile operating system that was invented and developed by Apple for its smartphones. It is used on devices such as iPhones, iPads, Apple Watch, and Apple TV. Apple is the best-selling OS in the United States of America on smartphones.

Advantages

- iOS offers a clean and user-friendly interface, making it simple for users to explore and control their devices.
- Apple firmly controls both the hardware and software of its devices, coming about in optimized execution and consistent user involvement.
- The App Store is curated, guaranteeing higher quality apps and a more secure environment for consumers.
- Apple ordinarily conveys software program overhauls to all upheld devices at the same time, guaranteeing clients have gotten to the most recent highlights and security patches.
- Apple provides a comprehensive ecosystem for all its devices to work together.
- Apple focuses more on user privacy by enforcing data encryptions on its devices and limiting data collection and ad tracking.

Disadvantages

- iOS offers less adaptability for customization compared to Android, with fewer alternatives for tweaking settings and personalizing the user interface.
- Apple's strict App Store arrangements can restrain certain apps or highlights that are accessible on other platforms.
- Apple gadgets tend to be more costly upfront compared to numerous Android users, constraining availability to some clients.
- To completely utilize Apple's environment, you will discover yourself intensely dependent on Apple services and subscriptions.
- Whereas the integration between Apple devices is consistent, it can be restricted if you lean toward blending and coordinating devices from distinctive producers.
- Although some Apple devices now have USB-C, there are older versions that still use proprietary connectors such as Lightning which are incompatible with a wide range of accessories and other electronics.

- Users who need extra storage but don't want to spend more for higher capacity may find Apple devices lack storage capacity.

Android

Android is a mobile operating system developed by a consortium of developers known as the Open Handset Alliance then acquired by Google in 2005. It is used in smartphones, tablets, Android TVs, cars, and watches. Android also is the best-selling OS in the world on smartphones.

Advantages

- Android offers a more extensive choice of devices than Apple which gives users the ability to find devices that fit their needs based on color, size, and shapes.
- Android users have the option to customize their devices in ways that iPhones lack. They can install different home screen launchers that give a personalized experience and run smoothly on their devices.
- Android provides the ability for users to control their apps, permissions, and processes that run in the background.
- A new iPhone is more expensive than an Android with similar specs.
- Android provides better file management where users can simply transfer files between devices and any computer with no setup required.
- Android is more open source compared to iOS, cultivating advancement and adaptability for designers and devotees to adjust the working framework and make special encounters.
- Android has more apps and games than Apple. Such features open doors to flexibility where users can easily install apps from any source, including third-party app stores.

Disadvantages

- Developers often complain about Android when creating apps because their ecosystem is fragmented which happens by having a variety of devices.
- Security on Android devices raises concerns among users due to the open nature of the platform and the diversity of devices and manufacturers which makes their devices more susceptible to malware and security vulnerability.
- Android software updates are usually delayed because of the variety of devices. There are too many devices from diverse manufacturers which results in a lack of uniformity in terms of feature availability and security patches.
- Android applications drain the battery faster than iPhones because some apps remain active in the background.

Mobile Device Platform Comparison Table

	Advantages	Disadvantages
Android	- Extensive customization options. - Diverse hardware choices.	- Fragmentation across devices. - Security concerns due to openness.

	• Seamless integration with Google services. • Open-source nature fosters innovation. • Range of affordability options.	• Inconsistent user experience. • Delayed software updates for some devices.
iOS	• Sleek and intuitive user experience. • Tight hardware-software integration. • High-quality curated App Store. • Timely software updates.	• Limited customization options. • Dependency on Apple services. • Higher upfront cost. • Restricted app ecosystem.

Mobile Device Platform Decision

Choosing Android over Apple can be profitable for our project for a few reasons. Android's advancement environment, based on Java, adjusts well to the abilities that the group now has, especially since Java is the essential object-oriented language instructed at UCF. This recognition assists the improvement preparation and diminishes the learning bend. In addition, the accessibility of an Android phone inside the group encourages simple testing and investigating of code, empowering effective improvement both within the working environment and domestically. Android's broad customization choices cater to fitting the client encounter to a particular extent of prerequisites, whereas its open source is prominent, more straightforward, and adaptable in debugging issues, giving the group the tools required to handle challenges successfully.

3.7.2 Framework

Framework Technology Comparison

A framework technology is a pre-built structure that is used to build software applications. It is used by developers as a template to address diverse types of applications or development challenges, such as web development, mobile app development, or machine learning. It consists of multiple resources such as libraries, tools, and other methods throughout the development process. Framework technology plays an important role in helping developers save time and energy, create clean and adaptable code, and reduce costs. There are several types of frameworks. Here are some that are suitable for our project.

Mobile App Framework

A mobile app framework is a software creation platform that is used by developers to create mobile applications. It includes different sets of tools and software, compilers, debugging tools, and programming interfaces. Even though mobile applications come in different types of shapes and sizes, Android, iOS, as well as Windows are among a few of the digital devices available. Based on how applications work on smartphones mobile apps can be classified into three categories.

Native Apps

They are designed for specific operating systems such as Android, iOS, and Windows. To make an application run on both Android and iOS, it is required to develop the app twice which makes the development process both slower and more expensive.

Web Apps

A Web app is software that can be accessed via a web browser interface and can't be installed on devices. With the variety of mobile device screens, developers need to be equipped with adaptive design and interface elements specifically for each device.

Hybrid Apps

Hybrid Apps can run on multiple platforms whether it's iOS or Android. On the other hand, they require complex interaction from the device which leads to an expensive cost to maintain the app for both multiple operating systems.

All three major frameworks have their upsides and downsides as we've outlined in the preceding sections. Our team chose native apps not only for using Android as platform-specific. It provides high-performance and full access to device features and superior user experience which is an important aspect. In the table shown on the next page, we summarize the advantages and disadvantages of each type of mobile framework.

Framework Technology Comparison Table

	Language	Dependency	Maintenance	Compatibility
Native App	Platform-specific	Low	Frequent updates required	Specific to each platform
Web App	Web Technologies	High	Easier to update, centralized	Universal across all devices with a web browser
Hybrid App	Combination of native and web technologies	Low	Regular updates needed	Broad, but with some platform-specific tweaks

Framework Product Comparison

Let's extend our research on mobile applications by looking into a few well-known cross-platform systems such as React Native, Flutter, Xamarin, and Ionic. These frameworks are built to develop mobile applications that can run on different platforms by giving adaptability and efficiency to designers.

React Native

React Native is a framework that was built and supported by Facebook. It gives developers access to create mobile applications for Android, iOS, web, and desktop using Java, C/C++, JavaScript, and React. Companies like Tesla, Airbnb, Skype, and Amazon Prime use React Native to develop their applications. React Native has an immense set of libraries, solutions, and is part of a larger developer community where developers offer support and share their experiences. These experiences inspire others like me to write better code with time. It ensures stability for mobile applications and code reusability. Developers often use React Native because of the hot reloading feature which makes it easier to develop and implement. In contrast, React Native lacks a user-friendly interface for projects that deploy on multiple screen sizes. Debugging is also an issue for developers where many programming languages are used in one project.

Flutter

Flutter is an open-source framework developed by Google to create native mobile applications for Android, iOS, web, and desktop. It is used by several companies including Google, Alibaba, Abbey Road Studios, and BMW. It has the hot load feature that reflects instantly without rebooting and an excellent performance where developers can write their code and deploy it across different platforms. Flutter uses the Dart programming language. Dart is known for its speed, performance, and scalability. In addition, it is a single language for Full-Stack Development where developers use it on both Front-end and Back-end. On the other hand, it lacks third-party libraries, maturity, and limited community support.

Xamarin

Xamarin is an open-source, cross-platform mobile app development framework building iOS, Android, and Windows with .NET. It was acquired by Microsoft in 2016. Olo, Rumble, Zappiti, UPS, Alaska Airlines, and BBVA are among the companies that used Xamarin to build their mobile applications. Using C#, Xamarin gives developers access to build a platform-specific UI for their projects. Unlike other platforms, Xamarin developers can participate in the code and fix bugs. Furthermore, it is considered the simplest for maintaining and updating purposes where any changes made in one platform reflect in another. The Xamarin third-party library is compatible and accessible for developers to use on other platforms. However, its licensing is expensive, and performance depends on the complexity of the project. Xamarin is strongly discouraged because it is set to discontinue its support and services from May 1, 2024.

Ionic

Ionic is a free and open-source framework to create mobile applications using web technologies like HTML, CSS, and JavaScript/TypeScript. It allows developers to build Android and iOS applications using a single codebase. Companies like Aflac, Target, Southwest Airlines, H&R BLOCK, IBM, and T-Mobile use Ionic to develop their applications. Ionic is used by many to build because it's easy to find resources to create Ionic apps without having to learn new skills. Using HTML and CSS makes Ionic a flexible framework for customizing. It has a set of libraries that include UI components, common app icons, and plugins. It also has some disadvantages. For

example, Ionic is not recommended for graphics-heavy applications and is not compatible with older devices. In addition, it required additional maintenance for third-party plugins.

Framework Product Comparison Table

	Language	**Performance**	**Deployment**	**UI Flexibility**
Flutter	Dart	Near native	iOS, Android, Web, Desktop (Beta)	High
React Native	JavaScript/ TypeScript	Near native	iOS, Android, Web	Moderate
Ionic	HTML/CSS/JS	Web view-based	iOS, Android, Web	Limited UI flexibility
Xamarin	C#/.NET	Native	iOS, Android, Windows, Mac	Moderate

Framework Product Decision

All four major platforms have their upsides and downsides as we've outlined in the preceding sections. In the table above, we summarize their advantages and disadvantages. In conclusion, React Native stands out for its robust performance, extensive ecosystem, and ease of adoption for developers familiar with JavaScript and React. Its ability to deliver near-native user experiences across platforms makes it a preferred choice for many mobile app development projects.

3.7.3 Database

Database Technology Comparison

A database management system is a software that is used by developers to manage, store, and organize data in a computer system. Modern applications use databases to collect data like usernames, email addresses, product colors, prices, and ratings. Rows and columns are used in databases to organize data by providing easy access to developers. There are several types of databases that can be classified into two categories such as SQL (relational) and NoSQL where each provides different advantages to developers. Here are some popular ones that are suitable for our project.

Hierarchical Databases (NoSQL)

Hierarchical databases were developed by IBM to store information. It's also known as the oldest database. Data in hierarchical databases are organized in a tree structure which makes it easy to add and delete information. It works fast with linear data storage and handles one-to-many relationships. On the other hand, it doesn't support many too many relationships. In addition, the search process is very slow because it requires a database management system (DBMS) to run through the entire model from top to bottom until the required information is found.

Network Databases (NoSQL)

Network databases are similar to hierarchical except data is organized in a graphical representation which implies multiple parent records. It is a simple database and easy to design. Unlike hierarchical databases, network databases support one-to-many and many-to-many relationships where complex data are accessible effectively. However, the use of pointers to maintain records makes the structure of the database very difficult for developers. Network databases require a large number of pointer adjustments to insert, delete, and update any records.

Relational Databases (SQL)

A relational database was developed by E. F. Codd in 1970. It's known as a relational database management system (RDBMS) where data is organized in rows and column structure using Structured Query Language (SQL) to create, read, update, and delete (CRUD) data. Oracle, MySQL, Microsoft SQL Server, PostgreSQL, and DB2 are examples of relational databases. Developers use relational databases for several reasons. It is the most popular and secure database model. It offers support to both data independence and structure independence concepts to design, maintain and administer the database. Moreover, its structure doesn't affect data access when changes are made and allows developers to write complex queries to access or change the data from the database. Conversely, developers find it very difficult to map objects in relational databases. It's also not suitable for large projects and is expensive.

Object-Oriented Databases (NoSQL)

An object-oriented database is a system that represents in the form of objects. It is a structured database that uses object-oriented programming concepts like inheritance, polymorphism, and overloading to store data. Object-oriented databases support different types of data. For example, pictures, voice, video, including text, numbers. Compared to other databases, object-oriented database code is cheaper to maintain. Its code is reusable. However, OODBMS is limited and lacks a theoretical foundation and security support.

Relational (SQL) vs. Non-Relational (NoSQL) Database Comparison Table

	Relational	Non-Relational
Scalability	Vertical and Horizontal	Horizontal Share data between servers
Data Structure	Organize data in rows and columns. Strict schema	Document-oriented Key-value Graph Wide column Store unstructured data
Use Cases	Complex Queries Transactions	Real-time applications, collaborative tools

Language	Structured Query Language (SQL)	Varies
Performance	Small projects	Support large projects. Instant access to users Unlimited data storage
Security	ACID Safe	Weak Limited ACID

Database Product Comparison

When it comes to popular database brands used by developers, Firebase Database, MongoDB, and MySQL stand out prominently. These platforms have different advantages that are suitable for several projects. Below, we examine the advantages and disadvantages of each choice that could make the design process more achievable.

MongoDB

MongoDB is a NoSQL document-oriented database. It uses a query language similar to JSON called MongoDB Query Language (MQL). MongoDB is part of the MERN stack, which consists of MongoDB, Express, React, and Node. This stack allows the developers to write codes in JavaScript on both the front end and back end of the application. Therefore, there is less programming language overhead when implementing the application. MongoDB is fast and easy to manage data. MongoDB is compatible with both SQL and NoSQL types which allows developers to easily combine it with different database management systems. It's designed for horizontal scalability to manage large amounts of data and high-performance use cases. However, effectively managing data poses a challenge because of the possibility of duplication. It does not support joins compared to relational databases and using it can slow down execution and affect performance.

MySQL

MySQL is one of the most popular relational database systems. It is an open-source relational database owned by Oracle. MySQL belongs to the LAMP application software which stands for Linux, Apache, MySQL, and Perl/PHP/Python stack. Knowing C and C++, it is a suitable database to use because it works well with such system platforms as Windows, Linux, and MacOS. It's also known as a simple and easy database to use because it only requires basic knowledge. As a business-oriented platform, it is easy to configure and supported by the most popular cloud providers such as Amazon, Microsoft, and others. It supports ACID (Atomicity, Consistency, Isolation, Durability) to ensure data validation and consistency across multiple operations for applications that handle sensitive or critical data. However, MySQL doesn't work well with large databases. Data is often corrupted because it lacks handling transactions. Finally, MySQL development and debugging tools are not efficient compared to other databases.

Firebase Database

Firebase is a NoSQL database that stores data in a JSON-like format owned by Google to develop web and mobile software. There are two types of Firebase databases. First, Firebase Realtime Database allows real-time access to data on multiple platforms. Last, Cloud Firestore is used to handle large and complex data models. Both Realtime and Firestore allow developers to easily access, manage, and update data even when there's no Internet connection. It is an easy-to-use environment to start a project that doesn't require too much software development expertise. In addition, it is part of a large community which makes it easy to find answers to problems that pop up. In contrast, Firebase is not compatible with other platforms which makes it difficult to migrate data. It is limited to search functions and has constraints to execute multiple queries simultaneously. Developers cannot use other cloud service platforms like DigitalOcean, AWS, or Azure because it is restricted to Google Cloud.

Database Product Comparison Table

	Advantages	Disadvantages
MongoDB	● Simple data access, storage, input, and retrieval ● Easy compatibility with other data models ● Horizontally scalable solution	● Extensive memory consumption ● Data insecurity ● Complicated process to interpret into other query languages
Firebase Database	● Beginner friendliness ● Convenient data access ● Faster App Development ● Integrated Machine Learning	● Limited querying capabilities ● Limited data migration ● Restricted to Google Cloud
MySQL	● Simple syntax and mild complexity ● Cloud compatibility ● Security	● Scalability challenges ● Partial open-source support ● Limited compliance with SQL standards

3.7.4 Code Editor

In the world of programming, choosing the right Integrated Development Environment (IDE) plays an important role for any developer. With so many options available, it can be challenging to decide which one best suits based on our needs. Eclipse, IntelliJ IDEA, and Visual Studio Code are the three most popular used by developers. Each of these popular IDEs has its advantages and disadvantages based on the developer's needs and preferences. In this section, we will explore the differences and summarize their strengths and weaknesses in the table below.

Visual Studio Code

Visual Studio Code is also known as VS Code and is not to be confused with Visual Studio. It is a code editor developed by Microsoft to build and debug modern web and cloud applications.

Advantages:

- Visual Studio Code is an open-source, fast, and lightweight IDE that can support large projects.
- It supports a variety of programming languages and many useful extensions and plugins.
- Visual Studio Code has a large community that allows developers to customize their IDE to their specific needs.
- Git support is fully integrated into Visual Studio Code which allows developers to manage version control directly within the IDE.

Disadvantages:

- Visual Studio Code lacks features and performance for larger enterprise-level projects.
- Customizing Visual Studio Code may require some learning experience with extensions and settings.

Eclipse

Eclipse is an open-source integrated development environment primarily used for Java and developed by the Eclipse Foundation.

Advantages:

- Eclipse has an extensive number of plugins available for developers to customize code according to their needs.
- It has a large community that consists of resources, tutorials, and support for developers.
- Eclipse is suitable and has a variety of features for Java developers because it was initially designed for Java development.

Disadvantages:

- Beginners find Eclipse difficult to use because its interface and configuration are overwhelming.
- Eclipse user interface is outdated compared to other IDEs.
- Eclipse lacks performance especially when working with large projects or installing many plugins.

IntelliJ IDEA

IntelliJ IDEA is a popular code editor developed by JetBrains that is primarily used for Java development and other programming languages such as Kotlin, Groovy, Scala, and Python.

Advantages:

- IntelliJ IDEA has an intelligent code completion feature that helps developers write code faster and with fewer errors.
- It has a wide set of built-in features for different programming languages.
- IntelliJ IDEA has a modern and intuitive user interface.

Disadvantages:

- IntelliJ IDEA has limited resources.
- Advanced features are only available in the paid Ultimate Edition, which may not be affordable for some developers or organizations.

Code Editor Comparison Table

	Advantages	Disadvantages
Visual Code Studio	- Lightweight and Fast - Extensive Language Support - Rich Extension Marketplace - Integrated Git Support	- Less Robust for Large Projects - Limited Built-in Features - Learning Curve for Customization
Eclipse	- Extensive Plugin Ecosystem - Community Support - Suitable for Java Development - Free and Open Source	- Steep Learning Curve - Performance - Outdated U
IntelliJ IDEA	- Smart Code Assistance - Rich Feature Set - Excellent Refactoring Tool - Modern User Interface	- Cost - Resource Intensive

3.7.5 Mobile Cloud Computing (MCC)

Mobile Cloud Computing is used to deliver mobile apps through cloud technology. It allows mobile devices to utilize cloud resources and services to perform tasks such as authentication, location-aware functions, and providing targeted content and communication for end users. This method which is a PaaS (Platform as a Service) provides access for developers to build, update, and deploy their mobile apps using cloud services from any device without having to worry about storage capacity, processing power, and battery life limitations. The Smart House Tinting app will require users to store personal information such as usernames and passwords. It will also have geographic access to the devices to perform real-time actions. Therefore, it is important to consider a cloud platform to host our mobile app.

Let's extend our research on Mobile Cloud Computing by diving into some popular platforms used by developers to host their mobile apps. In this section, we will explore their advantages and disadvantages and summarize them in the table below.

Heroku

Heroku is a container-based cloud founded in 2007 that enables developers to build, run, and operate applications entirely in the cloud. It supports a wide range of programming languages like Ruby, Node.js, Python, Java, PHP, and Go. Heroku can handle growing user bases. In

addition, it is part of a large community and has extensive documents available to assist developers. It also has a variety of add-ons that allow developers to extend the functionality of their applications with third-party services. On the other hand, Heroku has underlying infrastructure issues that limit control over server configurations. It can be expensive for large-scale applications or those with high traffic volume. Its performance may not be as high compared to other cloud platforms. Finally, Heroku is considered a vendor platform because it may be challenging to migrate to another platform in the future.

Microsoft Azure Mobile Apps

Microsoft Azure is one of the top cloud computing platforms developed by Microsoft. It is known for its extensive range of services, including virtual machines, databases, Artificial Intelligence, and analytics tools. It provides scalability for several levels of demands which handles increased traffic as user bases grow. This platform supports iOS, Android, and Windows which allows developers to build apps for different devices using a single backend infrastructure. Azure has other built-in services such as Azure Functions, Azure Notification Hubs, and Azure SQL Database for building and managing mobile applications. It is a secure platform with several features such as authentication, encryption, and access control to help protect data and infrastructure from unauthorized access and breaches. In contrast, Azure can be challenging for novice users to set up and configure and maintenance may require a certain level of expertise. Azure is a vendor lock-in platform. It offers a flexible price model which tends to increase as usage and demand increase.

AWS Amplify (Amazon Web Services)

Amazon Web Services, Inc. (AWS) is a subsidiary of Amazon founded in 2002. It is a cloud platform that provides scalable solutions services ranging from individuals, companies, and governments without the need for large upfront investments in hardware. AWS is a flexible platform for diverse business needs, from hosting simple websites to running complex machine learning algorithms. It helps businesses protect their data and meet regulatory requirements which include data encryption, identity, and access management (IAM). AWS is an affordable platform that offers pay-as-you-go pricing without the need for long-term contracts or upfront costs. Despite AWS's high availability, occasional outages can occur, impacting the accessibility of applications and services hosted on the platform. It is a vendor lock-in which is difficult to migrate to other platforms in the future.

Mobile Cloud Computing Platform Comparison Table

	Advantages	Disadvantages
Heroku	- Supports various programming languages. - Handles growing user bases well. - Extensive community and documentation.	- Limited control over server configurations. - Costly for large-scale or high-traffic apps. - Performance may lag compared to other platforms.

			● Vendor lock-in complicates future migration.
Microsoft Azure	● Wide range of services including AI and analytics. ● Scalability for growing user bases. ● Supports multiple platforms for app development. ● Secure platform with authentication and encryption.		● Complex setup and configuration require expertise. ● Vendor lock-in and potentially increasing costs.
AWS Amplify	● Scalable solutions without upfront hardware investments. ● Data protection features like encryption and IAM. ● Affordable pay-as-you-go pricing.		● Occasional outages impacting accessibility. ● Vendor lock-in, difficult migration to other platforms.

4. Design Constraints and Standards

In this section, we will discuss the various standards and constraints that we had to actively consider and that we were facing during the course of this project. The main objective of this section is to shine a spotlight on some specific standards and constraints that played a role in how we went about developing our project. We also delved into some of the rules and standardizations that our group had to be aware of as we developed the different parts of our project and how everything was going to be implemented.

4.1 Standards

Standards are present in all aspects of our everyday life, from the different technologies we use to the food we eat and the buildings we live in. Standards are documents that provide specific specifications and procedures designed to ensure the safety, reliability, and efficiency of the products, materials, and various things we use daily. One example of a standard that we use every day is Wi-Fi. Anyone with the ability to connect to Wi-Fi can access any Wi-Fi network, provided they are granted access and can connect and use that Wi-Fi network. This is due to the standard that has been established based on the IEEE family of standards. The following subsections will go over the different standards that went into developing KnightTint.

4.1.1 IEEE 802.11 Standard

The IEEE 802.11 standard, commonly referred to as Wi-Fi, is a foundational technology for wireless local area networks (WLANs) that has revolutionized the way devices connect to the internet and communicate with each other. Developed by the Institute of Electrical and Electronics Engineers (IEEE), this set of protocols specifies the means of implementing wireless network communications over short distances using radio waves. Since its initial ratification in

1997, the IEEE 802.11 standard has undergone numerous amendments to enhance its speed, security, and efficiency, accommodating the ever-growing demand for wireless connectivity in personal, commercial, and industrial settings.

At the heart of the IEEE 802.11 standard is the definition of the medium access control (MAC) and physical (PHY) layers that dictate how data is transmitted over wireless connections. The MAC layer is responsible for maintaining order in the use of shared media, ensuring that data packets are sent and received without collision through mechanisms like carrier sense multiple access with collision avoidance (CSMA/CA). The PHY layer, on the other hand, deals with the transmission and reception of signals over the air, including modulation, signal encoding, and managing transmission power.

One of the key features of the IEEE 802.11 standard is its support for various frequency bands, primarily the 2.4 GHz and 5 GHz bands, which allows for flexibility in network setup and operation. Over the years, amendments to the standard have introduced new functionalities and improvements. For example, IEEE 802.11a introduced higher throughput in the 5 GHz band, IEEE 802.11b offered improved range in the 2.4 GHz band, IEEE 802.11g combined the benefits of both, and IEEE 802.11n (Wi-Fi 4) further increased speed and reliability through the use of multiple-input multiple-output (MIMO) technology. More recent updates like IEEE 802.11ac (Wi-Fi 5) and IEEE 802.11ax (Wi-Fi 6) have continued this trend, significantly boosting wireless network performance to meet modern data transmission needs.

Security has also been a focal point of the IEEE 802.11 standard's evolution, with protocols such as WEP (Wired Equivalent Privacy) initially providing basic security, which was later found to be inadequate. This led to the development of more secure protocols like WPA (Wi-Fi Protected Access) and WPA2, and most recently, WPA3, which offer robust encryption and authentication methods to protect wireless networks from unauthorized access and data breaches.

In summary, the IEEE 802.11 standard has played a pivotal role in the development and widespread adoption of Wi-Fi technology, enabling a myriad of devices to connect wirelessly to networks and the internet. Its continuous evolution through amendments ensures that wireless networking technology keeps pace with the growing demands for higher speed, increased security, and greater efficiency in our increasingly connected world.

4.1.2 C Language Standard

Very similar to a lot of technology that we use around us every day, the knighting system requires the use of code to program some of the features that we plan on implementing. As we learned above, C is the best language when it comes to use in embedded IT systems and we plan on implementing these algorithms using the C language. Hence, understanding the international standard ISO/IEC 9899, which defines the C programming language, is crucial.

The standard is kept up by the International Organization for Standardization (ISO) and the American National Standards Institute (ANSI). The current version of the standard is an updated version of the previous version with not that many changes, with new changes to come in 2024. The standard covers different aspects of the C language as a whole. It covers the lexical elements which include the different identifiers, keywords, constants, and operators that are used

throughout the code. It also controls the different data types that are available and how they are handled when it comes to storage and other aspects. It covers the expressions and operators; this includes assigning operators, and the different arithmetic operators; all these things are defined within this standard.

This standard is crucial for the continuity of the C language across all the different platforms that are available. It makes sure that there is consistency and compatibility across all these different compilers and platforms that are out there. It also helps the people programming in C to write code that can run, be compiled, and executed in the same way regardless of the specific system or compiler that is currently being used. Making sure that all of our systems adhere to the standards will make it so that any code we write can run and compile on the system that will be implemented within KnightTint.

4.1.3 Battery Standards

Our battery choice, the Energizer L91, adheres to rigorous battery specification and testing standards. The most significant standard for this battery selection is ANSI C18.3M. ANSI is a private, non-profit organization that creates voluntary consensus regulations that help promote safety and global competitiveness of United States businesses. C18.3M covers the scope, purpose, and safety of portable lithium primary cells and batteries. ANSI was also the organization that created the overall AA standard.

The L91 batteries also adhere to the standards of the International Electrotechnical Commission. This organization publishes international standards for electrical, electronic, and related technologies. The L91 adheres to the IEC-FR14505 (FR6) designation, which is a standard for the voltage, weight, volume, and discharge of the lithium AA battery.

These batteries are certified by the testing and certification organization Baseefa for ATEX (Atmosphères Explosibles) compliance. ATEX is a European Union directive, which focuses on potentially explosive equipment. They are also tested by the certification company SGS for compliance with international health, safety, and performance standards. These two certifications help distinguish the Energizer L91 as a battery of utmost quality and safety.

The Energizer L91's compliance with such rigorous standards ensures its reliability and safety in a wide range of applications. This battery is designed to perform in extreme conditions, making it ideal for use in high-demand electronics such as digital cameras, GPS units, and emergency equipment. Its adherence to the ANSI and IEC standards not only underscores its quality but also its capability to deliver consistent performance. Furthermore, the ATEX certification signifies that the L91 can safely operate in environments with potentially explosive atmospheres, essential for industries such as mining, oil and gas, and chemical manufacturing. These certifications not only ensure the battery's robustness but also enhance its marketability across diverse sectors, fostering trust among consumers and professionals alike.

4.1.4 UL/CE Standards

UL (Underwriters Laboratories) and CE (Conformité Européenne) standards are integral in ensuring the safety, performance, and regulatory compliance of hardware components within a project. When developing hardware for the North American market, adherence to UL standards is paramount. These standards encompass a breadth of criteria, including electrical safety, fire resistance, mechanical integrity, and environmental impact. Products undergo rigorous testing procedures to verify their compliance with UL requirements. The UL mark, prominently displayed on certified products, serves as a trusted symbol of quality and safety assurance, instilling confidence in consumers, manufacturers, and regulatory authorities alike.

Hardware intended for distribution within the European Economic Area (EEA) must meet CE standards. CE marking signifies conformity with essential health, safety, and environmental protection requirements outlined in EU directives. Unlike UL certification, the CE marking is often self-declared by the manufacturer or importer based on their assessment of conformity with relevant EU directives.

In choosing hardware for this project, we look for signifiers that the product adheres to one or both of these sets of standards. This helps us ensure that we are compiling safe, high performance components into this project, which will in turn help us build a higher quality end product.

4.1.5 RS-485 Standard

The RS-485 standard, also known as TIA/EIA-485, is a widely recognized specification for serial data communication that facilitates the high-speed transmission of data over long distances in electrically noisy environments. Developed in the mid-1980s, this standard has become a fundamental communication protocol used in various industrial, commercial, and laboratory settings. The RS-485 standard supports multi-point systems, distinguishing it from its predecessor, RS-232, which is limited to point-to-point communication.

One of the key features of RS-485 is its differential signaling technique, which transmits data using two wires, often referred to as the 'A' and 'B' lines. This method involves sending a signal as the voltage difference between these two lines, significantly enhancing noise immunity and allowing reliable data transmission over distances of up to 4000 feet (approximately 1200 meters) at rates up to 10 Mbps, depending on the cable length and the environment.

RS-485 supports a network configuration known as a "bus" topology, where up to 32 devices (nodes) can be connected along a single pair of wires, without the need for a central controller. This capability can be extended to support even more devices with the use of repeaters. Each device on an RS-485 network can communicate with any other device on the network, provided they adhere to a common protocol, making it a versatile standard for systems requiring multiple data sources and destinations.

The electrical characteristics of RS-485 allow for both half-duplex and full-duplex communication. In a half-duplex configuration, devices share the same pair of wires to both transmit and receive data, but not simultaneously. In contrast, full-duplex communication uses an

additional pair of wires, allowing simultaneous data transmission and reception, which can significantly increase the efficiency of data exchange in certain applications.

The robustness of RS-485 makes it ideal for industrial environments where electrical noise from motors, generators, and other equipment can cause significant interference with sensitive electronic signals. Its ability to operate over long distances with minimal signal degradation also makes it suitable for building automation and networks that require reliable communication between devices spread over large areas.

Despite the emergence of more advanced communication standards like Ethernet and wireless technologies, RS-485 remains a popular choice for many applications due to its simplicity, reliability, and cost-effectiveness.

4.1.6 DMX512 Standard

DMX512 (Digital Multiplex 512) is a communication protocol developed by the United States Institute for Theatre Technology (USITT) in the 1980s and is commonly used in lighting and effects control. It is a specific implementation of the RS-485 protocol discussed in the previous section, giving it a more standard. DMX512 outlines the format for data transmission, including the structure of the data frame, start code, and channel values. The protocol's consistent data structure enhances interoperability between different devices, simplifying setup and integration.

DMX512 operates unidirectionally, where data flows from a controller to the lighting fixture or dimmers. The unidirectional nature of DMX512 simplifies the network topology and reduces the likelihood of data collisions, which is crucial in maintaining the integrity of the lighting effects in dynamic environments. Each DMX512 frame consists of a start code followed by up to 512 channels of data, hence its name. Each channel is capable of carrying an 8 bit value, representing attributes like intensity, color, or position. As it is based on RS-485, data is transmitted differentially over 2 wires. DMX512 uses a specific transmission rate of 250 kBaud, or 250 kbps.

Physically, DMX512 uses a 3 or 5 pin XLR connector, where two pins are the positive and negative of the differential data and a third is ground. Using standard XLR connectors makes the system compatible with many existing audio and video equipment setups, thereby reducing the need for special adapters. A simple adapter can also be used to run a DMX signal over an ethernet cord for short distances, which is relevant to the hardware of this project.

4.1.7 Microcontroller Communication Standards

In the realm of microcontroller technologies, communication protocols are a cornerstone, defining the rules and formats for data exchange between devices and systems. These standardized protocols not only facilitate interoperability but also ensure reliability and efficiency in data transmission, which is crucial across various applications, from simple sensor networks to complex industrial control systems.

Serial communication protocols like I2C (Inter-Integrated Circuit), SPI (Serial Peripheral Interface), and UART (Universal Asynchronous Receiver/Transmitter) are widely used in microcontroller-based projects. I2C, known for its simplicity and efficiency in connecting multiple slave devices to a master, uses only two wires for communication, making it ideal for

setting up internal communication between components on a single board. SPI, on the other hand, stands out for its higher data transfer rates, utilizing a master-slave architecture that can support multiple slaves with separate select lines, making it suitable for applications requiring rapid data transactions.

UART communication, characterized by its asynchronous transmission method, allows for direct serial communication between microcontrollers and peripheral devices or between two microcontrollers. This protocol is particularly useful in applications where data needs to be transferred over longer distances, or where only minimal wiring is desirable.

For applications requiring robustness in noisy environments or over longer distances, the RS-485 protocol is a standard choice. Its differential signaling enhances noise immunity, allowing reliable communication in industrial settings where electromagnetic interference is common. Moreover, RS-485's ability to support multi-point configurations makes it an excellent option for networked systems requiring multiple devices to communicate over a single pair of wires.

In the context of networked and IoT (Internet of Things) applications, protocols like MQTT (Message Queuing Telemetry Transport) and CoAP (Constrained Application Protocol) are gaining prominence. MQTT, designed for lightweight messaging between devices, is ideal for scenarios where bandwidth and battery power are limited. CoAP, on the other hand, is optimized for web-based communication in constrained environments, enabling efficient resource discovery and management.

The adoption of wireless communication standards such as Bluetooth, Wi-Fi, and Zigbee has further expanded the capabilities of microcontrollers, enabling them to connect and interact within the vast ecosystem of IoT devices. These standards ensure seamless integration and communication between a diverse array of devices, from home automation systems to industrial sensors, fostering the development of increasingly interconnected and smart environments.

The evolution and standardization of these communication protocols have been instrumental in the development and widespread adoption of microcontroller technologies. By providing a framework for reliable and efficient data exchange, these protocols empower engineers and developers to create innovative solutions that can communicate and operate seamlessly within a myriad of environments and applications.

4.1.8 IPC PCB Standards

The IPC, previously known as the Institute for Printed Circuits and now called the Association Connecting Electronics Industries, plays a pivotal role in establishing uniformity in the manufacturing and assembly processes of electronic equipment. This global entity, driven by its members from various sectors within the electronics industry including design, PCB manufacturing, assembly, and testing, is dedicated to promoting reliability and durability in electronic products. These efforts are critical in standardizing processes and materials used across international borders, ensuring that electronic products can be reliably produced and maintained worldwide. IPC's influence extends across all types of PCBs, be it single-layer,

double-layer, or multilayer boards, by setting a wide array of standards covering design guidelines, material specifications, and performance criteria among others.

For instance, the IPC-2220-FAM series represents a comprehensive set of design specifications for printed boards, encompassing several individual standards like IPC-2221, IPC-2222, and IPC-2223, each addressing different aspects of PCB design. The IPC-221B standard outlines the general design requirements applicable to various types of printed boards, supporting the more specific guidelines found in standards like IPC-2222B for rigid organic printed boards and IPC-2223 for the design of flexible printed boards. Moreover, IPC-2225 focuses on the design standards for Organic Multichip Modules (MCM-L) and their assemblies, taking into account various considerations such as thermal, electrical, and mechanical factors. This hierarchical structure of standards ensures that every facet of PCB design is thoroughly covered, promoting a holistic approach to quality and reliability in electronic manufacturing.

Adhering to IPC standards is crucial for ensuring that PCB designs meet the desired reliability and quality benchmarks, facilitating effective competition in the market. These standards not only enhance the end product's performance and longevity but also foster clear communication within the global electronics industry. By following IPC guidelines, designers and manufacturers can avoid misunderstandings, meet quality requirements efficiently, and ensure smooth production workflows.

4.1.9 Sensors Standards

Sensor standards are crucial guidelines that ensure reliability, compatibility, and efficiency in the design, manufacturing, and deployment of sensors across various industries. These standards cover a wide array of parameters including sensor performance, environmental conditions, data quality, and interoperability among different systems. The International Electrotechnical Commission (IEC), IEEE Standards Association, and the International Organization for Standardization (ISO) are some of the primary organizations that develop and maintain these standards. By adhering to established sensor standards, manufacturers can guarantee that their products perform consistently in diverse conditions and interface seamlessly with other electronic systems.

The importance of sensor standards is particularly evident in sectors like automotive, healthcare, and industrial automation, where precision and reliability are paramount. In the automotive industry, for instance, sensors must perform under extreme conditions and comply with standards such as ISO 26262 for functional safety. Healthcare uses standards such as ISO 13485 to ensure that sensors used in medical devices are safe and provide accurate readings critical for patient care. Industrial automation relies on standards like IEC 61131-2 and IEEE 1451 to harmonize the performance of sensors in complex machinery, ensuring they communicate effectively and operate efficiently in an integrated environment.

Future developments in sensor technology and the increasing application of IoT devices are driving the evolution of sensor standards. With the advent of smart cities, autonomous vehicles, and personalized healthcare, new standards are being developed to address the unique challenges of these technologies. These include standards for data privacy, cybersecurity, and new methodologies for sensor calibration and maintenance. The continuous updating of sensor

standards is essential not only for enhancing device interoperability and user safety but also for fostering innovation and technological advancement in an increasingly connected world.

4.1.10 Voltage Regulation Standards

When examining the application of voltage regulation standards in consumer electronics and appliances, these guidelines become instrumental in guaranteeing the safety, dependability, and functionality of devices used in daily life. Standards such as IEC 60950, outlining safety prerequisites for information technology equipment, hold particular relevance in the realm of consumer electronics. Complying with this standard assures that devices like smartphones, laptops, and routers feature robust voltage regulation mechanisms, shielding users from potential risks associated with overvoltage or undervoltage circumstances.

Furthermore, voltage regulation standards like IEEE 1159, focusing on monitoring power quality, are pivotal in upholding the dependability of consumer electrical systems. For instance, voltage fluctuations and harmonic distortions can adversely affect the performance of sensitive electronic equipment commonly present in households, such as TVs, gaming consoles, and home automation devices. Adhering to these standards empowers manufacturers to devise products capable of enduring variations in voltage levels and functioning optimally under diverse electrical conditions, thereby augmenting the overall user experience.

In the domain of renewable energy and smart home technologies, standards like IEEE 1547 and IEC 61850 are gaining prominence among consumers. With the increasing adoption of solar panels, wind turbines, and energy storage systems in households, ensuring compatibility with existing electrical grids and adhering to voltage regulation standards becomes imperative for seamless integration and peak performance. Likewise, smart home devices rely on steady voltage levels to operate efficiently and communicate with one another. Compliance with standards in communication protocols and power system interoperability guarantees that devices such as smart thermostats, lighting controls, and security cameras function reliably and effectively, contributing to energy conservation and heightened convenience for consumers.

Overall, voltage regulation standards emerge as vital safeguards for consumers and enhancers of the performance of electrical and electronic devices across various consumer applications. Through adherence to these standards, manufacturers can inspire trust in their products, mitigate risks linked with electrical faults, and furnish solutions that cater to the evolving needs and expectations of consumers in an increasingly interconnected and electrified world.

4.1.11 USB Standard

USB (Universal Serial Bus) is a widely used wired connection standard for connecting various devices to various devices. It can be used as an interface for data transfer, charging, and connecting peripherals. This connection standard has widespread user adoption and familiarity.

USB-A is one of the most widely used USB connector types. It features a rectangular shape with a flat, wider end, often found on the host side of a connection, such as on computers or wall-chargers. Micro USB and USB-C, on the other hand, are smaller and more compact

connector types designed primarily for portable devices like smartphones, tablets, and digital cameras. In our case, micro USB will be used to connect to power our development board. The standard electrical characteristics of USB is 5V with a rating of 500 mA, which is appropriate for a smaller device.

USB-C is used in this project for testing, where both the power and data signaling capabilities are utilized. This connector type has a data transfer rate of 10 Gbps, making it appropriate for relaying sensor data. The electronics market is shifting towards making USB-C the universal connector for all electronics, phasing out micro USB and proprietary connectors for a simpler and more convenient user experience. Overall, the USB standard is useful in maintaining compatibility and interoperability across a wide range of devices and manufacturers.

4.1.12 Outlet Standards

In the United States, the standard voltage for outlets is 110 or 120 VAC, 60 Hz. This voltage is distributed as a single-phase supply with three total wires, two hot wires carrying 120 volts each and one neutral wire. Standard power outlets in the U.S. are designed to accommodate plugs with two parallel blade prongs (hot and neutral) and a round grounding pin. This type of electrical outlet is defined by the National Electrical Manufacturers Association in the NEMA-15 standard. Adherence to this standard ensures the safe installation and use of electrical outlets across the country.

This standard is relevant to this project because we have to interface with outlets to power both our development board as well as our dimmable controller, as outlets are the only readily available source of constant high voltage. We must keep the electrical characteristics of the NEMA-15 standard outlet in mind while designing our hardware and the associated power distribution.

4.2 Constraints

4.2.1 Manufacturability Constraints

When it comes time to build our system there are a lot of different manufacturing constraints that we have to take into account. One of the main ones we have to take into account is the building of the demo house. The glass that we plan on using comes pre-cut at a certain size and don't plan on trying to cut it so to build the demo house we will have to make sure that it matches the size the windows come with and not try to make any significant changes. We are currently considering making the house out of wood but depending on the size of the demo pieces foam might end up being the more budget-friendly choice. We plan on also using some 3d printing to mount and secure together some of the housing parts to make sure that it looks decent for the demo. Additionally, careful planning will be required to ensure all elements are aesthetically aligned and structurally sound, to enhance the overall presentation quality of the demo.

Another constraint that we have to take into account when building our project is the fact that none of our group members have any background in mechanical engineering; most of our group have only worked with software and electrical stuff and have no experience with directly building something like this. One of our group members, while not having a lot of direct experience, has offered to work on it as a side project to get the demo house building ready to demo the project. This part of the project would be great when it comes to showing off the project and its different functions. It comes second to actually making sure that we achieve all of our set goals and making sure that our project works as described. This hands-on approach will not only provide valuable practical experience but also enhance team capability in dealing with physical system integrations.

4.2.2 Economic Constraints

Economic constraints when it comes to this project are going to vary from group to group depending on what the individual members are willing to do to get the project done. In some circumstances, our situation notwithstanding, there have been times when projects were able to get sponsored by outside companies so that they could save the team and also the company on the research and development that they would have had to do for a product that the team created. Because our project will be self-funded, we will need to make sure that we manage our budget properly and make sure we do the proper amount of research, and do the right comparison to make sure that we have the right and most important components of our system so that we are not going over budget on parts that we do not need. One example is something like an MCU; if we do not need it to be 16-bit, there is no need to spend the extra money to get a stronger MCU if the task we plan on using it for does not require that much power.

To conclude, the economics of this project is something that we will have to keep in mind at all times when we are designing this project. Choosing the best part for the job but also going above and beyond to make sure that it's not only the best part but also at the best price. This price-saving approach could also be seen as an advantage when it comes to other systems that might also try to do a system like this. Having the best parts at the best price will allow us to make a product that is available to everyone, the everyday homeowner, and not just business applications where the price is not always front of mind.

4.2.3 Sustainability Constraints

When designing this tinting system, we aimed to create a product that would last a long time and would not require significant maintenance or work to keep it functioning over time. Our vision is for the product to be so seamlessly integrated into your life that you don't even notice it's there, yet it's always performing its intended function. To ensure the user always has control over the system, we have implemented physical controls.

When considering the system as a whole, we identified that the window PCBs and the sensor that will be connected to them will be among our significant concerns. Being from Florida, our team is acutely aware of the frequent weather activity, such as rain or wind, that can damage windows and impair the efficiency of the sensor. Therefore, finding a way to mount the sensor where the user still has physical access, and where it can collect the necessary data to adjust the tinting levels accurately, will be one of our greatest challenges. We must also keep in mind the

temperature delta that exists in the different locations where the windows could be placed. For example,, in places like Florida, the outside temperature can reach levels above one hundred degrees, while in other states in North America, the temperature can drop below zero. Ensuring that the sensor and other components mounted to the window can withstand these temperature deltas is something that we will have to take into consideration when choosing our sensors.

Given the use case of our product, it is important that our group takes into consideration the long-term use of our product and how we ensure that the user experience does not degrade over time. To do this, we first have to address this on the software side, making sure that the application is updated, whether that's with new features or fixing bugs that might come up during the lifetime of the product. We cannot test for all scenarios, so making sure that we have a way to update our product if a problem does arise is a must for this project. Looking at this constraint, we also have to keep in mind the longevity on the hardware side. As we move into the digital age, a lot of products are doing away with physical controls and only implementing control via an app. What we see a lot of the time with these companies is that if they no longer exist or don't support the app anymore, the product then becomes useless and the user has to now get a whole new product. So, making sure that we have some form of physical control is something that we plan on implementing to ensure that the user always has control of their systems at any time.

4.2.3 Time Constraints

Of all the other constraints that our team is facing when it comes to this project, the time constraint will be one of the most important ones when it comes to how we deal with it. The first thing that we have to take into account is the fact that we are doing Spring and Summer rather than the usual Fall and Spring. This means that we will have a shorter span of time to work on getting our project going and having everything working on time. We will need to get everything started and tested early so that we can hit the ground running early so that we can test and get everything implemented as soon as possible.

One thing that we did as a team to try to combat this issue is to get a head start on getting everything started when it comes to this project. We started talking about putting a team together as early as the end of the Fall semester the year before. This allowed us to put some of the early time-consuming aspects of Senior Design 1, like getting your team together and getting to know each other, behind us early so that we could start working on the paper as soon as possible. This has for now allowed us to keep a head start on the other teams and we plan on maintaining it by keeping up with the writing and at least starting with some of our testing and prototyping early so that we don't run into any major issues when it comes time for senior design 2. This is also helpful in the case that a major issue does come up so we can address it without ending up falling behind on our work.

Another factor that affects our time commitment to this project is other prior commitments that we have outside of class time. Many of us in the group are taking other classes, some of us more than one, and some even have jobs that they have to keep up with so making sure that all of the group members are organized and making sure that our management of time is perfect is going to be crucial to making sure that we get the project done on time. Some of us also run into the issue of getting together into group meetings where we all have to commute for some distance to be

able to get on campus so making sure that we plan for that and account for everyone's ability to attend is something that we will have to keep in mind to make sure that our group is working in sync.

Lastly, one thing we have to take into account when it comes to time is the parts and the amount of time it will take from when we order them to when we get them so that we can start putting them together. For example, the PCB is often one of the main parts for previous projects with people finding issues with the different parts and at times having to replace the order for those parts. To avoid this being an issue we plan on ordering everything as early as the end of the current semester. That way we can have a heads-up if parts are not working and things are not arriving in a reasonable amount of time. We have also heard of different supply chain issues when it comes to companies like Texas Instruments who make a lot of MCUs that are planned to be used for this project. Making sure that these parts are ordered early is crucial to ensure the successful completion of this project.

4.2.4 Testing/Presentation Constraints

When we conceived our idea, we knew that our product would require a very specific situation to test and present some of the features we plan on implementing. One thing we considered is that, given it's a window, our system will need to be implemented within a house so that we can test the product's functionality and the benefits it can bring. Given how difficult it is to implement this system within an actual house, we decided early on as a group that we would design the system within a model home. This would allow us to do some of the testing we need to see the data on temperature changes and how the tinting system affects it.It would also serve as a great demo device if we were to replicate a lighting system, like having a heat lamp represent the sun, and demonstrate how the system is connected and performing its intended function.

We also took into account the fact that no one in our group has a background in mechanical engineering. This meant that building a demo house to showcase our product and its features would require us to step outside our comfort zone. The main constraint when it comes to the size of the house will depend on the size of the glass we can obtain from our manufacturing partner. Given the delicate nature of the glass system, we plan not to cut the glass to fit, but rather to build a demo that can easily accommodate whatever the size of the windows ends up being. This interdisciplinary challenge also underscores the importance of collaboration and learning in our project, pushing us to integrate knowledge from fields such as architecture and materials science into our design process. Furthermore, involving experts in these areas or consulting with structural engineers will provide crucial insights that enhance the structural integrity and functionality of our model home, ensuring it effectively demonstrates our product's capabilities.

For the final presentation, the demo construction must be built to demonstrate some of the transition and temperature features that we mentioned in the previous section. This means we might have to temperature-seal the entire house system to the best of our ability to ensure that the values we are getting match some of the benefits we predicted. Some features, like having multiple windows sync together, might be very difficult to demonstrate. This means we might need to have a video outside where we can get as much sunlight as possible to demonstrate some of the feature sets.

4.2.5 Environmental Constraints

The Smart House Tinting project aims to revolutionize building comfort, energy efficiency, and occupant well-being by allowing consumers to adjust window tint levels based on environmental conditions or personal preferences. While offering benefits like glare reduction, privacy control, and thermal management, the project must navigate significant environmental constraints throughout its design, development, and deployment phases.

In our endeavor, reliance on sensors for accurate light detection and tint adjustment is paramount. However, environmental factors, such as Florida's fluctuating natural light and weather conditions, pose challenges to sensor performance and calibration. To address this, our team is developing algorithms to calibrate sensors effectively, ensuring accurate measurements despite changing conditions. By minimizing light pollution and optimizing light distribution, we enhance comfort and energy efficiency while mitigating glare.

Temperature fluctuations also pose challenges, impacting the durability and energy consumption of project materials. In summer, our tinting solution reduces reliance on air conditioning by blocking solar heat transfer through windows and doors, maintaining a comfortable indoor environment. Conversely, in winter, it preserves indoor warmth, enhancing comfort year-round. To withstand extreme temperature changes, we prioritize materials and designs with robust insulation and thermal management techniques, ensuring sustainability and longevity.

Moreover, adhering to resource-efficient coding standards set by the United States Department of Energy not only minimizes resource consumption but also yields significant health benefits. By reducing UV radiation penetration, our tinting solution protects occupants from sunburn, premature skin aging, and skin cancer risks. Additionally, it safeguards materials from fading and deterioration, prolonging their lifespan. In summary, navigating environmental constraints is essential for the success of the Smart House Tinting project. By addressing these challenges through innovative technologies and sustainable practices, we aim to deliver enhanced comfort, energy efficiency, and occupant well-being in buildings.

4.2.6 Health and Safety Constraints

Window tint is not just a cosmetic upgrade or an economic strategy to help regulate the temperature inside to reduce the need for excessive air conditioning and improve overall energy efficiency. It can also have significant health and safety benefits as well. Our goal is not only to design a smart tinting window that works but does not harm people. As engineers, we are taking our product's health and safety constraints into great consideration by looking into codes and standards available to ensure people are safe and aware of the risks of using our product.

UV radiation from the sun can pose significant risks to human health and can also cause damage to materials. Many individuals enjoy basking in the natural light by standing or sitting next to windows to admire the scenic views. Our team is dedicated to ensuring that this enjoyment comes without any adverse effects, such as sunburn or an elevated risk of skin cancer. Through Smart Tinting technology, our mobile app empowers consumers to effortlessly adjust tint levels, effectively blocking harmful UV rays while still allowing pleasant visible light to filter through. Moreover, our focus remains steadfast on selecting window coatings that prioritize health and well-being.

The use of glass in our product introduces a crucial safety consideration, as mishandling or accidental drops can lead to its fragmentation into sharp pieces, potentially causing cuts and injuries to individuals nearby. To mitigate this risk, we select glass crafted from resilient materials renowned for their durability and safety properties. Moreover, our design prioritizes glass compositions with advanced temperature regulation capabilities, ensuring optimal conditions for stored items and averting the potential biohazard risks associated with inadequate temperature control.

Utilizing batteries instead of relying on electricity from a wall outlet to power our product offers unparalleled flexibility. However, ensuring that they pose no threat to health or safety is paramount. Given the potential for batteries to explode if mishandled, it is crucial for consumers to understand the risks involved. To address these concerns, our team employs a variety of strategies. Firstly, we exclusively utilize high-quality AA batteries that adhere to stringent safety standards, boasting features such as leak resistance and overcharge protection. Additionally, in our design process, we position the batteries in secure locations to minimize the likelihood of physical damage and heat buildup. This approach not only safeguards against moisture and contaminants but also bolsters the overall safety of the batteries.

5. ChatGPT Comparison with other Similar Platforms

In this section, we compare the different large language models currently available for public use and how they can be useful in helping us write our paper. In the current landscape, the three main players in large language models for public use is ChatGPT, created by OpenAI and released for public use in November of 2022, which has managed to grow significantly since then. Another player in the market is Microsoft Copilot, powered by a version of GPT-4 from OpenAI. Lastly, another competitor in the market is Google's own Gemini, formerly known as Bard, which plugs into Google's ecosystem of applications.

ChatGPT is particularly noted for its ability to engage in human-like dialogue, making it an excellent tool for brainstorming sessions and refining complex ideas into more understandable text. Microsoft Copilot enhances productivity by facilitating the integration of AI-driven content suggestions directly within Microsoft Office tools, streamlining the research and writing processes. Meanwhile, Google's Gemini leverages the extensive Google search database to provide factually accurate information and diverse perspectives, crucial for comprehensive and well-rounded academic writing

5.1 ChatGPT and Copilot

In the realm of artificial intelligence (AI), ChatGPT and Copilot stand as two prominent examples of how AI can help users in various tasks, from generating text to giving code suggestions. Both use ultramodern natural language processing (NLP) techniques to understand and respond to user inputs effectively. However, each platform has unique strengths and capabilities that cater to distinct user needs.

ChatGPT, developed by OpenAI, excels in generating human-like responses to textual prompts. Trained on a vast corpus of internet text, ChatGPT demonstrates impressive language understanding and coherence, enabling it to engage in meaningful conversations, answer

questions, and provide relevant information. Its performance is characterized by fluency, coherence, and contextual understanding, making it suitable for a wide range of conversational applications, including customer support, language learning, and entertainment. ChatGPT offers a user-friendly interface for interacting with the AI model through text inputs. Users can easily engage in conversations or pose queries, receiving prompt and coherent responses in return. Its simplicity and accessibility make it suitable for individuals without technical ability, facilitating seamless communication with AI-powered assistants for everyday tasks.

On the other hand, Copilot, a collaboration between GitHub and OpenAI, focuses on helping developers in writing code. Leveraging the power of GPT (Generative Pre-trained Transformer) architecture, Copilot analyzes code snippets and gives contextually relevant suggestions, including code completion, function definitions, and error handling. Its performance is evaluated based on the accuracy of code suggestions, speed of inference, and adaptability to various programming languages and frameworks. Copilot's ability to understand code semantics and generate syntactically correct suggestions enhances developer productivity and facilitates collaborative coding environments. Copilot integrates directly into code editors, providing real-time suggestions as developers write code. Its usability is optimized for software development workflows, offering features like auto-completion, documentation lookup, and code refactoring tools. Developers can leverage Copilot's capabilities to streamline their coding process, reduce errors, and accelerate project development, enhancing overall productivity and code quality.

To conduct the following comparison, I used both ChatGPT and Copilot free versions, giving them the prompt: "I would like to create a mobile app for a smart tinting window project. Please compare and select a list of software technologies to use. Provide at least three options." Upon comprehensive evaluation of both ChatGPT and Copilot platforms in response to the provided prompt, a nuanced analysis unfolds, highlighting the strengths and nuances of each. While both platforms are undeniably useful, it's imperative to approach their outputs with a degree of skepticism, as their information may require verification due to occasional inaccuracies.
Initially, it is discernible that ChatGPT exhibits a propensity for rendering outputs with a higher degree of accuracy when juxtaposed with Copilot. This superiority in accuracy can be attributed to ChatGPT's extensive training data and refined algorithms, which enable it to generate content that closely aligns with the input prompt. Conversely, Copilot presents a compelling proposition by augmenting its outputs with multiple sources, thereby enriching the generated content with a diverse array of perspectives and information. This multifaceted approach not only enhances the comprehensiveness of Copilot's responses but also instills a sense of credibility and depth in its outputs. By drawing from a myriad of sources, Copilot demonstrates a capacity to offer a more holistic view of the given prompt, incorporating various viewpoints and supporting evidence into its generated content.

Because Copilot integrates current content from the internet, it often has information on topics that ChatGPT is unfamiliar with, especially when asking about hyper-specific topics. When asked about the battery standard ANSI C18.3M, ChatGPT reported that it did not recognize the standard. However, when the same information was requested from Copilot, the latter gave a comprehensive overview of the standard. A similar issue occurred when information on the IEC-FR14505 designation. ChatGPT gave a breadth of information about the wrong standard

and when further questioned, reported that it did not recognize the requested designation. Copilot, on the other hand, recognized the standard and gave an overview. This not only demonstrated Copilot's broader knowledge base, but also the need to be diligent in using these generative AI models, as they often output their best guess without communicating their level of confidence or reliability. This is especially important when doing research on specific items in a broader catalog, such as standards.

Assessing the accuracy of ChatGPT is a nuanced endeavor. While the model demonstrates impressive proficiency in generating coherent responses, its performance can vary depending on the complexity of the task and the quality of the input data. In controlled environments, ChatGPT can achieve high levels of accuracy, rivaling human performance in certain language tasks. However, its susceptibility to generating nonsensical or contextually inappropriate responses underscores the need for caution when relying on its outputs. Continuous refinement and fine-tuning are essential to enhancing ChatGPT's accuracy and reliability over time. For the prompt given, ChatGPT output seems to be accurate even though some outputs are incorrect, and it doesn't provide sources to check the output. It's crucial to verify information independently to ensure reliability, as the model's responses might not always be entirely accurate or comprehensive. Furthermore, it is pertinent to consider the contextual relevance and coherence of the outputs produced by each platform. While ChatGPT excels in delivering more accurate responses, it is imperative to ensure that the generated content remains coherent and contextually appropriate. Copilot, with its ability to integrate multiple sources seamlessly, shows a commendable proficiency in maintaining coherence while enriching the content with diverse perspectives.

In essence, the comparison between ChatGPT and Copilot transcends a simple evaluation of accuracy; it underscores the nuanced interplay between precision, comprehensiveness, and coherence in natural language generation. While ChatGPT shines in its precision, Copilot's incorporation of multiple sources elevates the richness and depth of its outputs. Ultimately, I favor the output from ChatGPT, despite its incompleteness, over that of Copilot for this particular prompt. However, this preference is not without reservation, as both platforms offer valuable insights that merit consideration and verification in practical application.

5.2 ChatGPT and Gemini

When looking to start this paper there are several resources that are available out there that are able to help with looking for resources and checking on the information that we gather. In the past year the use of large language models in our everyday life has grown extensively and as these technologies grow our understanding of their capabilities has to also grow to determine how they will best be used to our advantage. For example, in my part of this project, I was responsible for the embedded section of the project and in my journey through the next paragraph, I would compare the usefulness of these different LLMs and how they can be helpful when making a project like this.

To start, we prompted Copilot Chat GPT with an information-based question. We asked it to compare the different programming languages that are available and are commonly used for embedded software applications. The result we got was pretty interesting. We found that Copilot was very good when it came to gathering the different information that's available out there.

When getting into a project like this, it's good to have a framework as to what you need to be looking for. What I found with the result is that Copilot was able to come up with a list of up to 8 different programming languages and their advantages and disadvantages. When looking at the specific information that it provides on the different languages, most of it appears to be correct, disregarding some of the more niche languages that it mentions. Another advantage that Copilot provides is its ability to cite the different sources of information that drove its response. This will help to better check the responses to make sure that they are accurate but also as a way to find the different sources that talk about your topic specifically. One thing that I do see being a potential issue is the LLM's selection of information when it comes to citing the information. Some of the articles that were cited were from well-known and trusted sources while others were from random sites that we have no way of verifying the accuracy of the information. This is something that we are going to have to keep in mind to make sure that the information that it's providing is not just coming from untrustworthy sources.

To further test its capabilities, I proceeded to prompt it to generate a piece of code for an MSP430 embedded board that would flash an LED to Morse code the word 'Hello'. When looking at the code that is generated, it looks to be very efficient, following all the rules for how code should be written when it comes to an embedded system board. What I do see is that Chat GPT took the approach of hardcoding the word 'hello' into the code and letting that be the only word that we can do. If the word needs to be changed, the whole first section of the code will have to be changed, which is very inefficient and wastes time. The code is also very well organized with comments and functions to make sense of the different steps that are involved in the problem. All in all, it's a great system when looking to get some assistance with embedded software and hardware.

On the other hand, to compare the two LLM capabilities, I proceeded to prompt it with the same question from the previous section and compare the differences in the questions. For the information-based question, I found that a lot of the information that was provided was very similar if not exactly the same. When looking at the result, Gemini takes a different approach to verify the information that is generated compared to Chat GPT. Gemini takes the approach of highlighting information that it finds might be argumentative and might need verifying, allowing the user to double-check and make sure that the information that is written is true, while Chat GPT just provides the source directly and leaves it to the user to go and do their own search on the information to make sure that it's right. I find that I like Gemini's approach a lot more when it comes to the early information gathering stages of the project because of how much information is available and out there, it is close to impossible at this moment to get the information and verify its credibility. So giving the user a heads-up to make sure that information is checked is a way of making sure that we are not allowing the LLM to hallucinate.

As previously mentioned, the same coding question was prompted and what I found in the testing is that the code was able to be generated a lot faster than the other LLM. While the code was a lot faster when going over it, I found that it was a lot more difficult to understand with a lot of functions all over the place to perform a task that was not difficult in the first place. While functions are great for making sure that the information is generalized, they can sometimes make the code difficult to read and understand when used excessively. With that being said, I find that the code generated was better with it being more generalized, allowing me to change the word to

something completely different very easily, which is great for future proofing. This LLM is a great resource to help with getting some idea as to the steps that are needed when trying to accomplish a task using these microcontrollers.

5.3 ChatGPT and Llama 2

Llama 2 is Meta's latest foray into the world of Learning Language Models. There are minor architectural differences. Compared to GPT-3, LLaMA uses SwiGLU activation function instead of GeLU, uses rotary positional embeddings instead of absolute positional embedding and uses root-mean-squared layer-normalization instead of standard layer-normalization. They have various different models for Llama, each trained by a different dataset and thus produces different responses. LLaMA's developers focused their effort on scaling the model's performance by increasing the volume of training data, rather than the number of parameters, reasoning that the dominating cost for LLMs is from doing inference on the trained model rather than the computational cost of the training process. Llama 2 foundational models were trained on a data set with 2 trillion tokens. This data set was curated to remove Web sites that often disclose personal data of people. It also upsamples sources considered trustworthy. Llama 2 - Chat was additionally fine-tuned on 27,540 prompt-response pairs created for this project, which performed better than larger but lower-quality third-party datasets. For AI alignment, reinforcement learning with human feedback (RLHF) was used with a combination of 1,418,091 Meta examples and seven smaller datasets.

Upon examining both Llama 2 and ChatGPT's capabilities regarding temperature sensors, we observed Llama 2's ability to deliver comprehensive and distinct information. It provided seven temperature sensor options: Thermocouples, RTDs, Thermistors, IR, Semiconductors, Capacitive, and Pyroelectric, each detailed individually. In contrast, ChatGPT offered five options but organized the data by parameter, which is advantageous for comparing specific characteristics across sensors. The device-centric approach of Llama 2 may be attributed to its specialized training and fine-tuning, which seems to facilitate a more user-friendly interface for tasks that necessitate a thorough comparison of devices. This arrangement allows for an integrated view of all pertinent details per device, enhancing usability for those needing a broad understanding of available technologies.

The distinct data presentation styles of Llama 2 and ChatGPT underscore the impact of architectural choices and training strategies on the utility and functionality of language models. Llama 2's design seems particularly advantageous for users requiring detailed, device-specific information, such as in educational or technical settings. Conversely, ChatGPT's parameter-focused output may better serve analytical purposes, where a direct comparison of specific features across various devices is essential. This divergence highlights the potential for models to be tailored to different tasks, enhancing their applicability and effectiveness in various contexts.

Moreover, these differences emphasize the importance of targeted model training and customization in developing practical AI tools. By aligning the training approach and architecture to the intended application, developers can enhance the model's relevance and efficiency in specific scenarios. This customization not only improves user experience but also optimizes resource utilization, making AI a more accessible and effective tool across diverse

industries and use cases. The evolution of language models like Llama 2 and ChatGPT reflects a growing understanding of how nuanced adjustments in model training and design can significantly influence their performance and suitability for specific tasks.

The comparative analysis between Llama 2 and ChatGPT also highlights the broader implications of strategic model development in the AI industry. As AI technology continues to advance, the focus shifts from merely achieving technological feats to fine-tuning these systems to better meet user needs and operational contexts. The choice of activation functions, positional embeddings, and layer-normalization techniques, although seemingly minor, plays a crucial role in shaping the model's behavior and output. These architectural nuances contribute significantly to the model's ability to process and present information in ways that align with specific user expectations and requirements. Such precision in development ensures that AI tools are not only powerful but also versatile and adaptable to varying professional and personal environments.

Furthermore, the commitment to creating more data-efficient models, as seen in Llama 2, suggests a trend towards more sustainable AI practices. By optimizing models to perform better with less computational demand during inference, developers are acknowledging and addressing the environmental and economic impacts of large-scale AI deployments. This approach not only enhances the accessibility of AI technology by reducing the operational costs but also mitigates the carbon footprint associated with running these complex models.

5.4 ChatGPT and Falcon

Falcon is an innovative suite of language models developed by the Technology Innovation Institute, marking a significant advancement in the field of generative AI. The Falcon family includes models of various sizes, with Falcon 180B being one of the most notable for its state-of-the-art performance. This model is a scaled-up version of its predecessors, like Falcon 40B, incorporating advancements such as multiquery attention to enhance scalability. Trained on an immense 3.5 trillion tokens using up to 4096 GPUs, Falcon 180B has been rigorously fine-tuned on diverse data, including web content, technical papers, and conversational datasets, to refine its capabilities in generating human-like text.

Falcon models, including the Falcon 40B-Instruct variant, are renowned for their high performance on various benchmarks and are accessible through HuggingFace Spaces for interactive engagement. The Falcon 40B-Instruct, in particular, has been fine-tuned on conversational data to facilitate smooth and coherent interactions, showcasing its proficiency in handling queries effectively. The integration of Falcon with tools like LangChain further expands its application possibilities, enabling developers to seamlessly incorporate this powerful AI into their projects. LangChain's HuggingFacePipeline is specifically tailored for models within the HuggingFace ecosystem, making the integration process straightforward and efficient. I prompted both ChatGPT and Falcon about different ways of smart tinting glass. ChatGPT gave me the regular slew of different tinting technologies we have presented in our research. I was unable to get a response from Falcon, as the program was incapable of responding to the prompt and would cut off the response before giving me any useful information.

As evidenced by the distinct performance contrast between ChatGPT and Falcon in handling specific queries, it becomes clear that even among advanced AI language models, functional differences in response mechanisms can be significant. This differential behavior could stem

from a variety of foundational elements in how each model processes and generates its responses. For instance, Falcon's inability to complete the response on smart tinting glass might indicate a limitation in its response generation algorithm or a possible glitch in its execution pipeline. This issue highlights the challenges in developing highly complex AI systems where even slight inconsistencies in training data, algorithmic structure, or resource allocation can dramatically affect output quality.

The incident also underscores the importance of robust testing and maintenance frameworks for AI models. Developers and researchers at the Technology Innovation Institute might need to revisit Falcon's training protocols and system architecture to diagnose and rectify the underlying issue that led to the incomplete responses. Continuous improvement efforts such as these are crucial in evolving AI technology to meet diverse and dynamic market needs. Moreover, such instances provide valuable lessons in the practical implementation of AI solutions, offering insights that go beyond theoretical constructs to address real-world application challenges effectively. These lessons are essential for refining AI models to ensure reliability and efficiency in user interactions.

6. Hardware Design

This section consists of detailed descriptions of KnightTint's hardware and construction. In this section, we start off with the design of all the hardware components connecting to each other such as the PCB and Development board to all of the sensors and peripherals and then the section goes further and displays a 3D representation of the model house used to demonstrate our project.

6.1 Component Package Decisions

We opted for 0805 component sizes in my PCB design due to their practicality and spatial efficiency. These components are sufficiently large for manual handling, which is particularly beneficial during the prototyping phase. The 0805 format also simplifies the soldering process, significantly reducing the risk of assembly errors. Additionally, there is a broad range of available electronic components in this size, such as resistors, capacitors, and inductors, allowing for greater design flexibility without compromising on functionality or component availability.

The choice was further influenced by the superior thermal and electrical performance of the 0805 components compared to smaller sizes like 0603 or 0402. These components are capable of managing higher power and dissipating heat more efficiently, thus improving the overall durability and performance of the PCB in scenarios involving elevated power or frequency. Moreover, the 0805 size offers a favorable balance between component cost and ease of use, providing cost-effective solutions for large volume orders. This makes the 0805 size particularly appealing for both exploratory projects and mass production, where maintaining high reliability and cost efficiency is crucial.

6.2 Voltage Regulation

Our PCB design will include a voltage regulator noise reduction and voltage consistency. All of our window PCB hardware is designed for 3.3V voltage input. However, the nominal voltage of

our batteries is 3.7V. We will utilize TLV70033DDC low dropout linear regulator to step this down to a constant 3.3V. This is a small regulator capable of accepting an input of 2V to 5.5V

We want to input a steady 3.3V for several reasons. Firstly, we want to make sure the voltage stays within the acceptable input range. If there is a spike in voltage for any reason, such as a faulty battery, we want to make sure that the hardware does not get damaged. A steady voltage input also helps minimize the impact of noise and interference on the circuit. Sudden changes or fluctuations in voltage can introduce unwanted noise into the system, potentially causing malfunctions or errors in sensitive electronic components. By using a regulator, we also can reduce the susceptibility of the circuit to external noise sources.

6.3 PCB Connection Diagram

The figure below displays the pin connection within and from the custom PCB. This gives a sub-system view of our hardware and shows the basic idea of how components are connected. The preceding MCU Pin Key below helps explain the function of each MCU connection to the other devices within the connection diagram.

The temperature and light sensor are connected via I2C, while the Wi-Fi processor and RS485 transceiver are connected via UART. We show the basic voltage regulation plan and power connections to our battery compartment and outlet. We also show the communication connections from our transceiver to the Gauzy controller and the subsequent connection to the PDLC. This diagram does not show the RLC components that are involved in these connections.

Microcontroller I/O Key

Pin	Description	Pin	Description
P1.4	General-purpose digital I/O	P2.2	General-purpose digital I/O
P1.5	General-purpose digital I/O	P3.0	General-purpose digital I/O
P1.6	USCI_B0: I2C data (I2C mode)	P3.1	USCI_B1: I 2C data (I2C mode)
P1.7	USCI_B0: I2C clock (I2C mode)	P3.2	USCI_B0: I2C clock (I2C mode)
P2.0	USCI_A0: Transmit data (UART mode)	P3.4	USCI_A1: Transmit data (UART mode)
P2.1	USCI_A0: Receive data (UART mode)	P3.5	USCI_A1: Receive data (UART mode)

Window PCB Connection Diagram
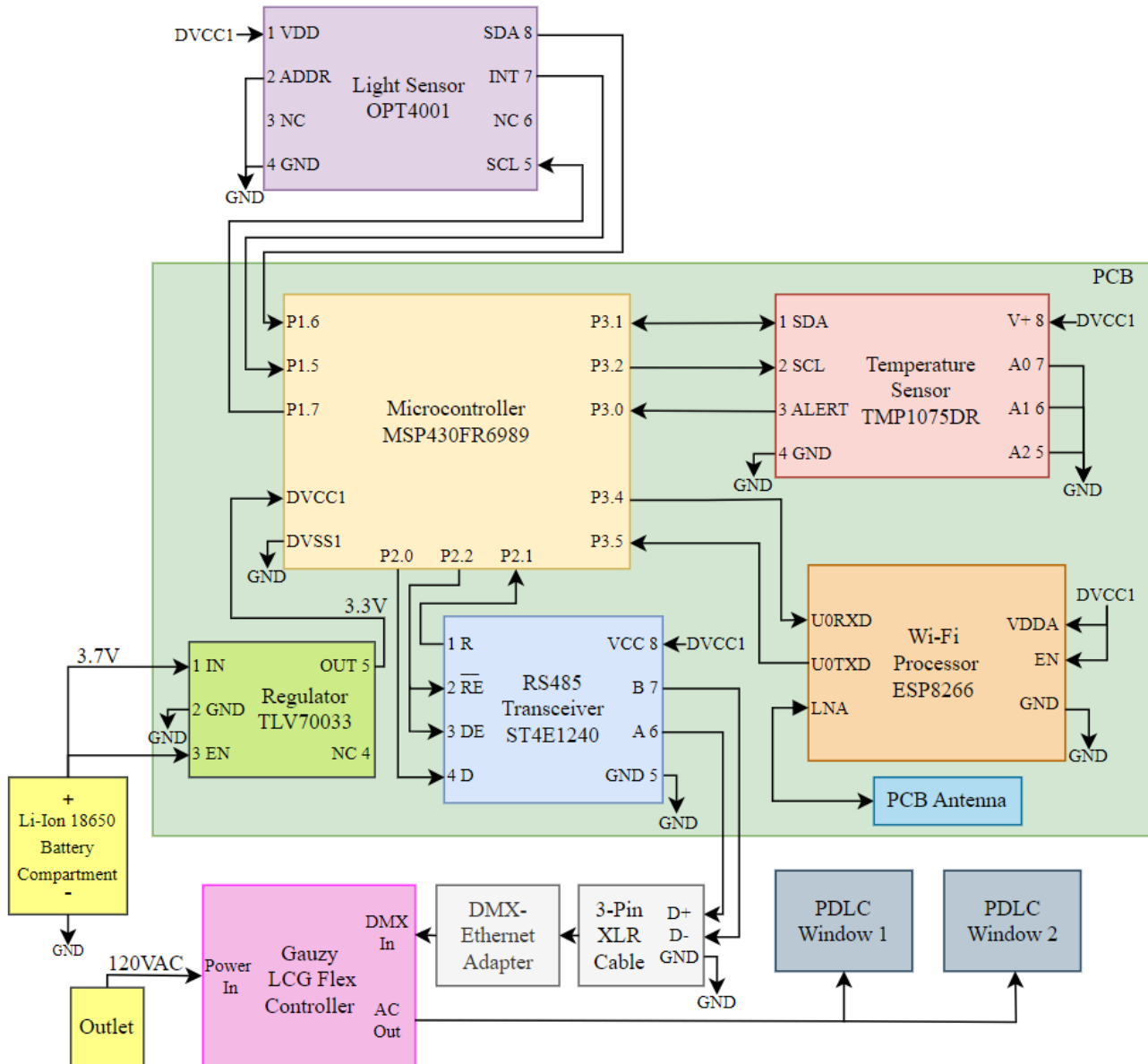
6.4 Power Requirements

Understanding the power requirements and distribution within our project is critical. Each component needs to be powered correctly for the system to function optimally. Without the proper voltage and current, our components won't work, affecting the entire system's performance as every element draws power. These components are the MCU development kit, the microcontroller, the light sensor, the temperature sensor, and the transceiver. The development kit will be running in its 12mW mode. This means that it will be requiring a voltage

of 5V for operation, and could consume up to 275µA for the board. The light sensor, temperature sensor and transceiver will require 3.3V for operation. Finally, the microcontroller will require 3.3V, and will consume an active-mode current of 240-275uA. The full power requirements are listed in the table below.

Power Requirements Table

	Voltage Requirement	**Power Supply**	**Active Current Requirement**
MSP430FR6989IPZ	1.8-3.6V	Battery-LR 3.7V to 3.3V	240-275 µA
OPT4001DTSR Light Sensor	1.6-3.6V	MCU VCC 3.3V	22-30 µA
TMP1075DR Temperature Sensor	3.0-3.6V	MCU VCC 3.3V	2.7 µA
ST4E1240DT RS-485 Transceiver	3.0-5.5V	MCU VCC 3.3V	1.77mA
ESP8266EX	2.5-3.6V	MCU VCC 3.3V	80 mA
MSP-EXP430FR6989 Launchpad Development Kit	1.8-3.6V	5V to 3.3V eZ-FET USB	240-275 µA
CC3100BOOST	3.3V	Dev Kit VCC 3.3V	~250 mA

6.5 Schematics

The following section contains the Altium Designer schematics for each hardware component, as well as the combined system. The schematics include pin connections, as well and RLC components necessary for system stability and design requirements.

6.5.1 Temperature Sensor

This is the schematic for the connections between our MSP430FR6989 microcontroller and the TMP1075DR temperature sensor. The V+ pin is connected to the 3.3V rail from the MCU DVCC1 pin. This input voltage has a bypass capacitor to reduce high frequency noise. The GND pin is wired to the PCB ground. SCL is the I2C clock input pin on the sensor and it has a connection to P3.2 of the MCU, which outputs this signal when in I2C mode. Similarly, the SDA pin, which is the I2C serial data pin, has a bidirectional connection to P3.1, the I2C data pin of

the MCU. The ALERT pin outputs a signal when the temperature passes a programmable threshold. It is wired to output to P3.0, which is a general purpose I/O pin. The SDA, SCL, and ALERT pins are wired with 5k pull-up resistors, which ensures that the voltage logic input remains a constant high when not being driven low. The 10nF capacitor and 5k resistors were chosen in accordance with the typical application circuit provided in the datasheet.
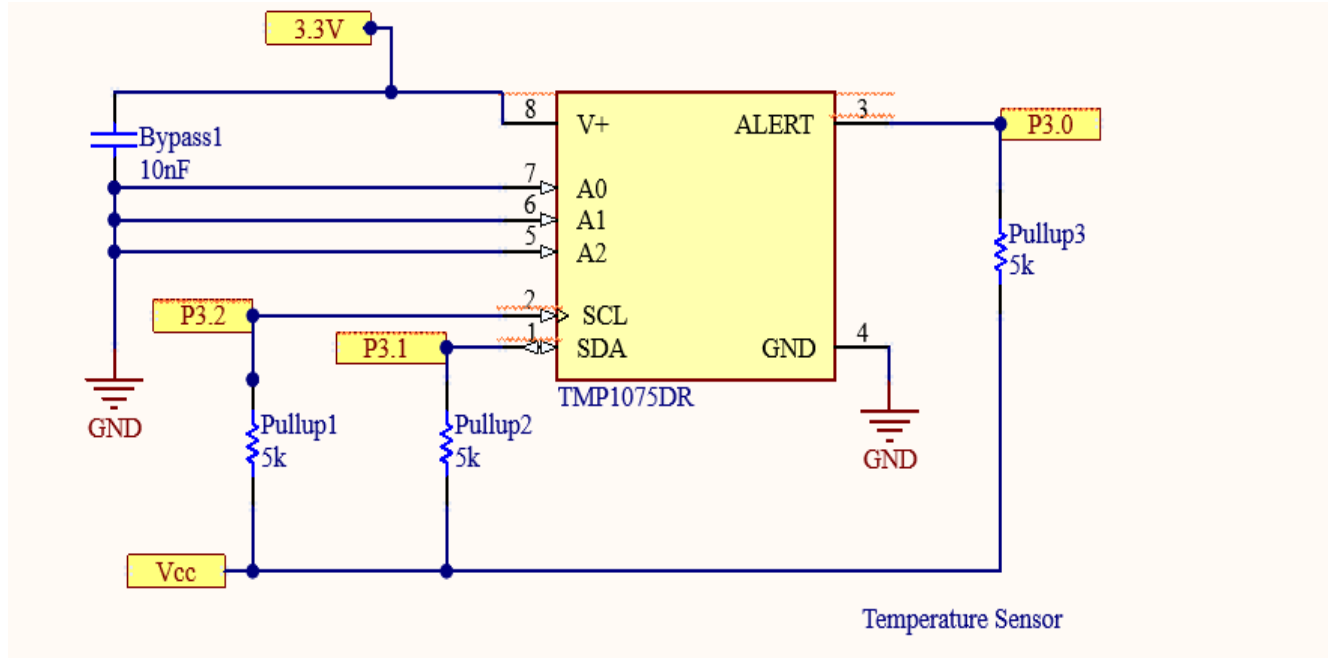

6.5.2 Light Sensor

The schematic below shows connections between the MSP430FR6989 microcontroller and OPT4001 light sensor. The VDD pin is connected to the 3.3V rail from the MCU DVCC1 pin. The GND pin is connected to the PCB ground. The 3.3V line has a 100nF bypass capacitor to ground to reduce high frequency noise. This sensor utilizes I2C communication, with the SCL and SDA pins connected to the I2C clock and data pins of the microcontroller, (1.7, and 1.6) respectively. The INT pin outputs an interrupt when a programmable lux threshold is crossed to P1.5, which is a GPIO pin. The SCL, SDA, and INT pins are connected to 10k pull-up resistors as shown in the datasheet's typical application circuit. This setup ensures reliable I2C communication and efficient handling of interrupt signals, crucial for real-time light sensing and response actions in embedded applications.

88
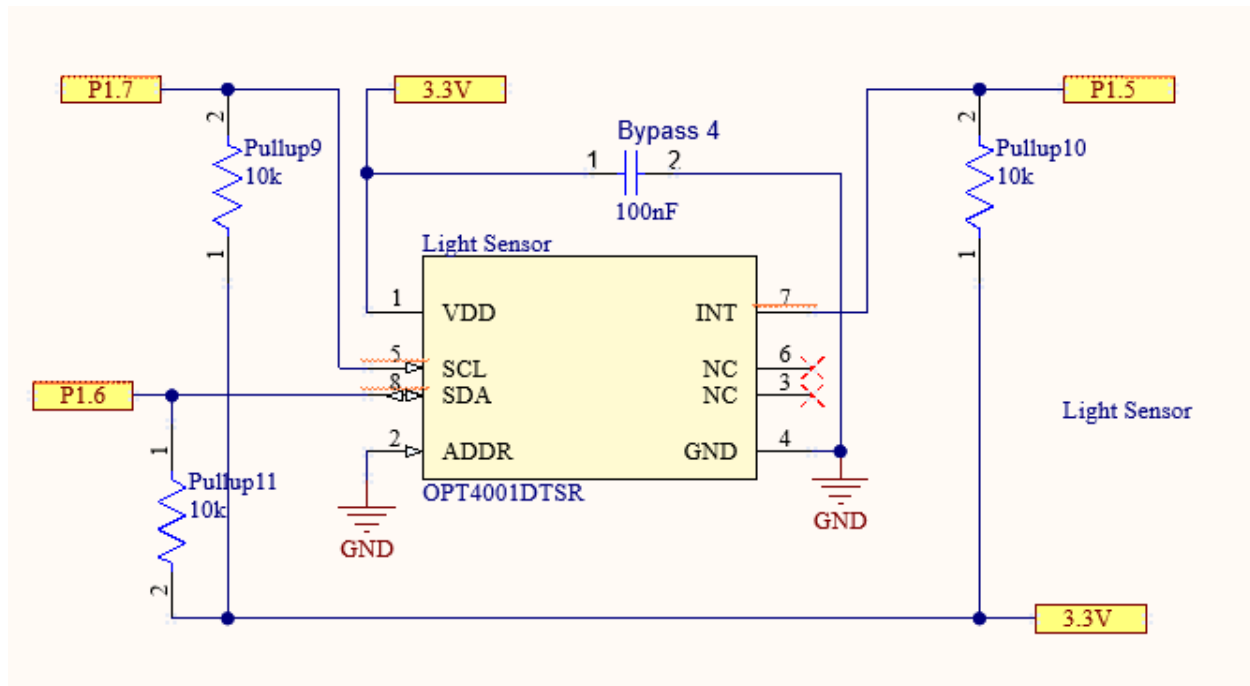
6.5.3 Battery and Linear Regulator

The schematic below shows our plan to implement the 18650 lithium-ion battery cells and convert this voltage into the 3.3V supply that we need for our PCB hardware. The three voltage sources represent three 18650 cells in parallel, which supplies 3.7V and extends battery life beyond one cell. This will be held in a 3x1 battery compartment. The TLV70033 DDCR is a low drop-out linear voltage regulator that outputs 3.3V from the OUT pin. The IN pin accepts an input of 3.7V from our batteries. Both of these pins are connected to ground by 1uF capacitors for better stability and transient performance. These values are recommended by the datasheet. EN is the chip enable, and is connected to a 5k pull-up resistor to maintain a constant logic level. The GND pin is connected to the PCB ground.

Battery and voltage regulator
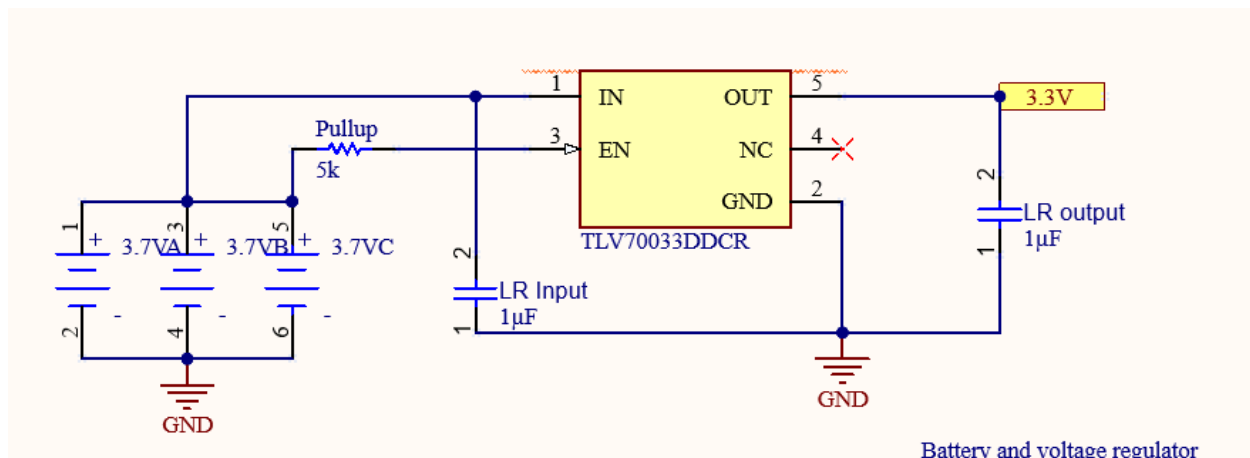
6.5.4 Wi-Fi Module

The schematic below shows our plan to implement the ESP8266EX Wi-Fi module. The power for this component comes from the 3.3V DVCC1 pin of the MSP430FR6989, which is connected to the power pins of the ESP8266 with a 1uF bypass capacitor to ground for noise reduction. The 3.3V rail is also connected through a pull up resistor to the CHIP_PU pin, which enables the pin and keeps a constant logic value. The RES12K pin is connected to a 12k resistor, which goes to ground according to the hardware guide for this chip. The U0RXD is the receiver pin for the UART communication and has a direct connection to the transmit pin of the MSP430. The U0TXD pin is the transmission pin for UART and is connected to the receive pin of the MSP430 through a 499 Ohm resistor following the hardware design guidelines. The GND pin is connected to the PCB ground. The LNA pin is the input pin for an antenna. It supports a variety of antenna types, but we have chosen a surface mount design antenna from existing successful implementations, the MOLEX 479480001. It has one SMD pad and three "dummy" SMD pads that are to be soldered to the board. The pi-network capacitors and inductor are values given in the hardware design guide for impedance matching.
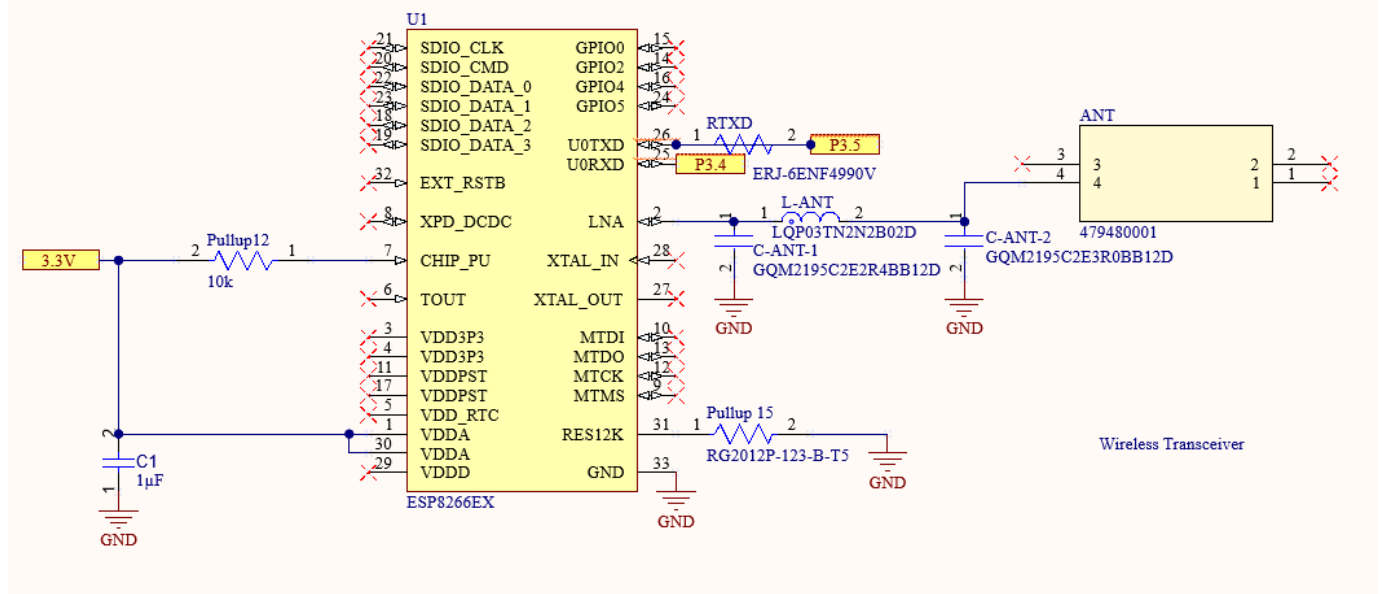
6.5.5 RS485 Transceiver

The schematic below shows the implementation of the RS-485 transceiver, which will allow us to interface our microcontroller with the Gauzy dimmable controller. The VCC pin of the transceiver is connected to the DVCC1 pin of the MCU with a bypass capacitor for stability. The RE' and DE pins are connected to the same GPIO pin of the MCU to toggle between send and receive operational modes. The R, which is the data output from the transceiver, is connected to the UART receive data pin of the MCU, and the MCU UART transmit data pin is connected to the receiver D pin, the data input. These unidirectional connections have a low pass filter according to their direction. The R and C values for these low pass filters were chosen to satisfy the equation $R*C = 0.032 / f_s$, where fs is the transmission rate (250 kbaud for DMX512). The A and B pins, which are the + and - data pins for DMX512, are connected to the + and - pins of a
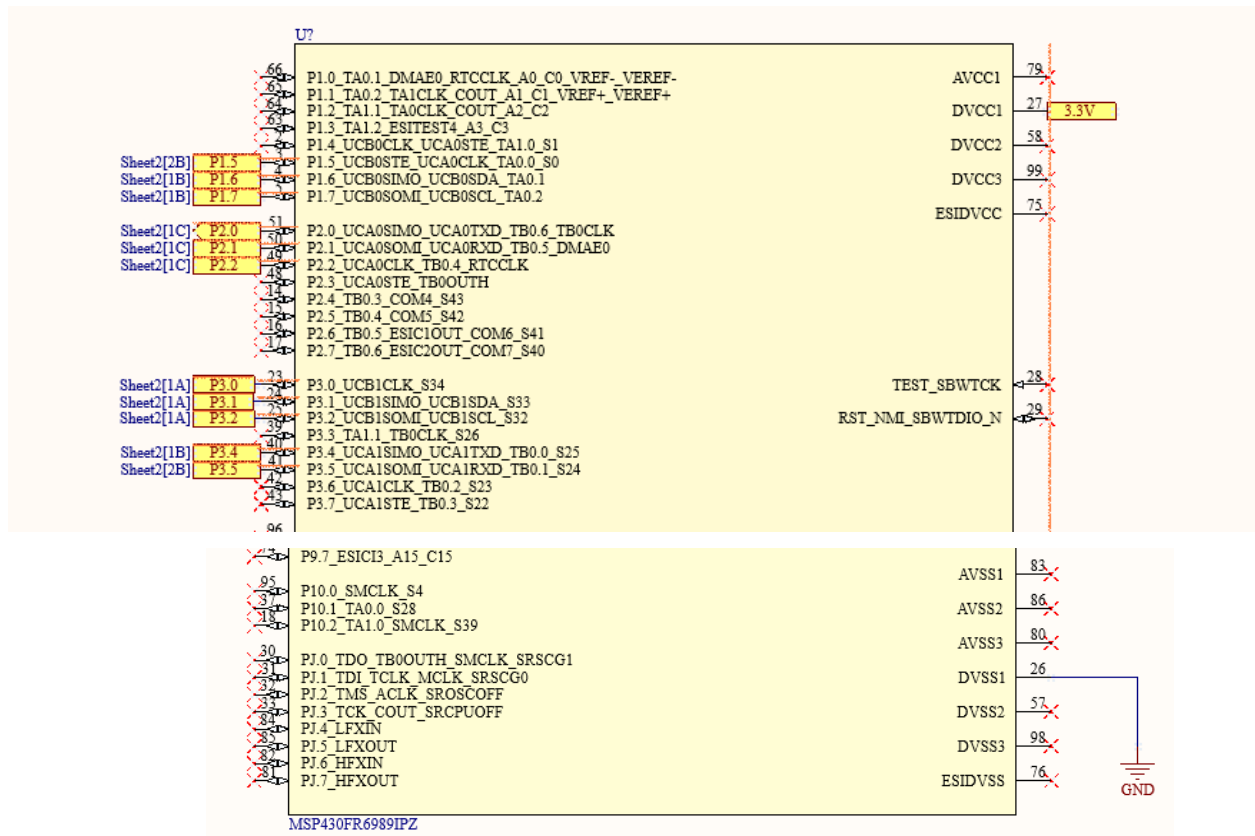
DMX cable connector. There is a 120 ohm termination resistor connecting the A pin to the B pin in order to match the impedance of the DMX cable. The ground pins of the connector are connected to the PCB ground.
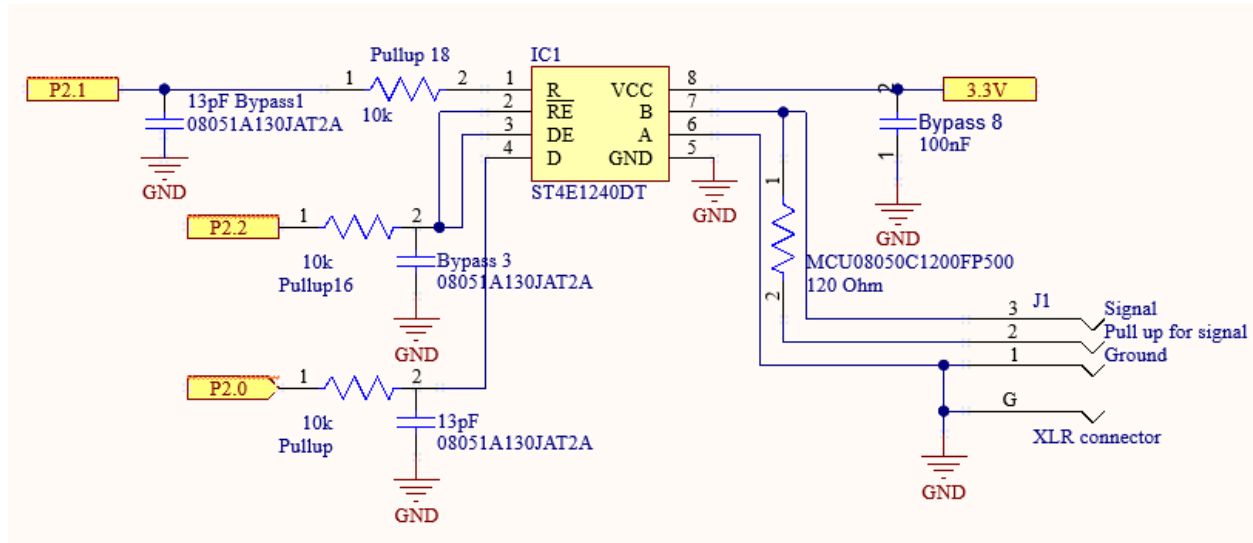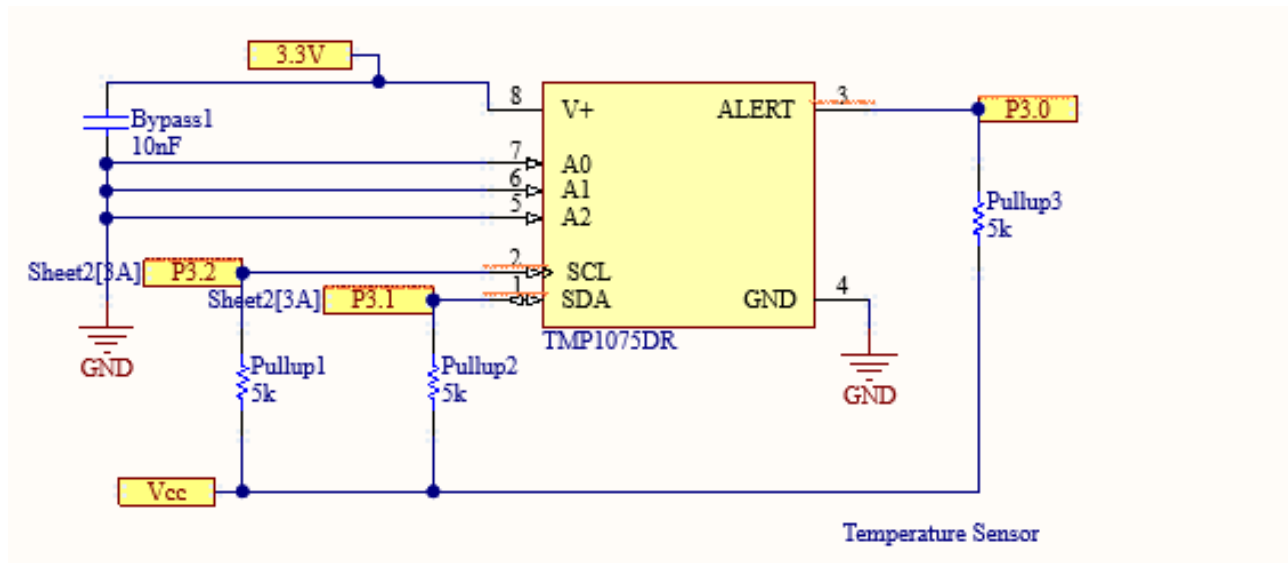

6.5.6 MSP430FR6989

Below is the schematic for the implementation of our MSP430 chip. This chip will enable us to control everything on the board, including reading and transmitting sensor information and adjusting the tinting values for our automatic mode. Ports 1.5 to 1.7 are connected to the SDA interrupt and SCL line for the light sensor, allowing data to be transferred to and from the sensor. Similarly, ports 3.0 to 3.2 are connected in the same way, but for our temperature sensor. The ESP chip is responsible for sending and receiving data from our main board and is connected using a UART connection at ports 3.4 and 3.5. In addition, the transceiver responsible for communication between MSP430 and the Gauzy controller is connected using a UART connection at ports 2.0 to 2.2. All of these connections allow the chip to serve as an intermediary between the sensors and the central hardware we plan to use.

Note, the schematic below has been cropped to only show the used pins of the MCU.

MSP430FR6989IPZ

6.5.7 Overall Schematic

Temperature Sensor

Battery and voltage regulator

Wireless Transceiver

Light Sensor

P2.1 — 13pF Bypass1 — 08051A130JAT2A — GND

Pullup 18 — 10k

IC1 — ST4E1240DT
1 R
2 RE
3 DE
4 D
8 VCC
7 B
6 A
5 GND
GND

3.3V
Bypass 8 — 100nF — GND

P2.2 — 10k Pullup16 — Bypass 3 — 08051A130JAT2A — GND

MCU08050C1200FP500 — 120 Ohm

J1
3 Signal
2 Pull up for signal
1 Ground

P2.0 — 10k Pullup — 13pF — 08051A130JAT2A — GND

G — XLR connector — GND

U?
66 P1.0_TA0.1_DMAE0_RTCCLK_A0_C0_VREF-_VEREF-
65 P1.1_TA0.2_TA1CLK_COUT_A1_C1_VREF+_VEREF+
64 P1.2_TA1.1_TA0CLK_COUT_A2_C2
63 P1.3_TA1.2_ESITEST4_A3_C3
62 P1.4_UCB0CLK_UCA0STE_TA1.0_S1
Sheet2[2B] P1.5 — P1.5_UCB0STE_UCA0CLK_TA0.0_S0
Sheet2[1B] P1.6 — P1.6_UCB0SIMO_UCB0SDA_TA0.1
Sheet2[1B] P1.7 — P1.7_UCB0SOMI_UCB0SCL_TA0.2
Sheet2[1C] P2.0 — 51 P2.0_UCA0SIMO_UCA0TXD_TB0.6_TB0CLK
Sheet2[1C] P2.1 — 50 P2.1_UCA0SOMI_UCA0RXD_TB0.5_DMAE0
Sheet2[1C] P2.2 — 49 P2.2_UCA0CLK_TB0.4_RTCCLK
P2.3_UCA0STE_TB0OUTH
14 P2.4_TB0.3_COM4_S43
13 P2.5_TB0.4_COM5_S42
16 P2.6_TB0.5_ESIC1OUT_COM6_S41
P2.7_TB0.6_ESIC2OUT_COM7_S40
Sheet2[1A] P3.0 — 23 P3.0_UCB1CLK_S34
Sheet2[1A] P3.1 — P3.1_UCB1SIMO_UCB1SDA_S33
Sheet2[1A] P3.2 — P3.2_UCB1SOMI_UCB1SCL_S32
P3.3_TA1.1_TB0CLK_S26
Sheet2[1B] P3.4 — P3.4_UCA1SIMO_UCA1TXD_TB0.0_S25
Sheet2[2B] P3.5 — P3.5_UCA1SOMI_UCA1RXD_TB0.1_S24
P3.6_UCA1CLK_TB0.2_S23
P3.7_UCA1STE_TB0.3_S22

AVCC1 79
DVCC1 27 — 3.3V
DVCC2 58
DVCC3 99
ESIDVCC 75
TEST_SBWTCK 28
RST_NMI_SBWTDIO_N 29

6.6 Non-Custom Hardware

This section includes our plan to power and integrate hardware that we are not designing ourselves. These pieces of hardware will be purchased and connected to our custom hardware in order to satisfy the functionality requirements of the system.

MSP-EXP430FR6989 Launchpad

This board is a pre-fabricated board from Texas Instruments that has the same MSP430FR6989 architecture as our custom window PCBs. We will be using this board as a central control board that relays instructions to the window PCBs. As we already satisfy the significant PCB design requirement of this project, we have chosen this board as to reduce the impact of time constraints. This board's peripherals, such as buttons and LEDs will facilitate manual control and testing. It will be powered using its eZ-FET 5V micro USB port, which will run from a simple

USB adapter box in a nearby wall outlet. This port also regulates the voltage down from 5V to3.3V. For 3.3 V, we will connect one of the development kit's jumpers across the J101 3V3 terminal.

CC3100BOOST

The CC3100BOOST is a Wi-Fi booster pack for the development kit above. It slots very easily onto the launchpad's pins and has its own 2.4 GHz antenna. The development kit will connect to this booster pack with its UART module to send and receive Wi-Fi signals from our window boards' ESP8266EX Wi-Fi modules.

Gauzy LCG Flex

The Gauzy LCG Flex dimmable controller is the controller sold by the company that manufactures the PDLC glass that we are using for this project. It outputs a square wave with a voltage that varies up to 70V. It will communicate with our control PCB at each window via a DMX cable. The connector for this cable is shown in the RS-485 transceiver schematic and the PCB layout. This DMX cable will be run through a DMX-Ethernet adapter, and the ethernet cable will plug directly into the "DMX IN" port of the Gauzy controller. The square wave output will be run to the PDLC glass through cables that contain 24 AWG conductors.

Dimmable Controller

For future designs we would like to implement our own dimmable controller. The dimmable controller consists of several pieces we aim to implement. Building our own dimmable controller would be an incredible boost to our project and cement our understanding of electronics. Our design will consist of a DAC to take in input from the microcontroller. That DAC will then be controlled through I2C to output an analog signal at different stages. This input will be entering into a high voltage op-amp that is non-inverting. The op-amp will be powered through a purchased 120V to 60V transformer.

7. Software Design

7.1 MSP430FR6989 Software and Components

For programming the MCUs we plan on using Code Composer Studio (CCS) which is an application made by Texas Instruments as an integrated development environment. CCS is made specifically for Texas instrument processors such as the module that we are using in this project. We plan on developing a program that has different modes where one mode is going to be classified as Automated. In this mode the program will collect the data from the sensors and will constantly check the values to detect any significant changes in the values. We also plan on having a manual mode where on the windows board themselves we will be able to control the tinting level of the glass to adjust it to the desired value. Other features like the schedule mode will be integrated within the app where the user can set a time for the tint to turn on and off.

7.1.1 MSP430FR6989 and CC3100 Development

To develop the program for the MSP430FR6989 and the CC3100 we use the correct development kits followed by the correct booster pack. The CC3100 is a Wi-Fi network processor from Texas Instrument which minimizes the host MCU, in this case our MSP430FR6089 software requirements. This booster pack enables Wi-Fi connectivity for specific embedded applications. This development kit accompanied by the booster pack enables easy development, troubleshooting, and flash of the MSP microcontroller. It also provides an LED and a screen on the board for use in this project which can be used to indicate the transfer and receiving of data. To connect the CC3100BOOST we will first need to attach it to the booster pins.. Then we can use a micro USB cable to flash the algorithm onto both of them.

7.1.2 Light Sensor

This device will be connected to our board that is going to be on the window and it will serve as a way to provide the automatic mode function to our controller. As part of the algorithm that will be written the information for the sensor will be taken in and based on the amount of light that is coming in the program will be able to adjust to different levels of tinting. The OP4001 supports the transmission protocol for standard mode (up to 100 kHz), fast mode (up to 400 kHz), and high-speed mode (up to 2.6 MHz) both the MCU and the eUSCI module must support the speed to be able to run the sensor at that speed. The eUSCI module in our MCU supports only the Standard Mode and the Fast Mode and so we will limit the I2C clock frequency to 400 KHz. After configuring our module then we will configure the system to start talking to the sensor. Once configured, the system will seamlessly communicate with the sensor, allowing for real-time adjustments based on incoming light levels, thus providing efficient and adaptive tinting functionality for enhanced user comfort and energy efficiency.

7.1.3 Temperature Sensor

This sensor, similar to the light sensor, will also be attached to the board on the window side and will provide us with information on how much light is currently present at that window. It will also determine if increasing or decreasing the intensity of the tint will affect the current temperature of the room. As part of the algorithm, we will have a set temperature that the user will set. Then we will take the information that is coming from the sensor and compare it to what the user has set, and change the tinting level based on that. This will be done using an SPI module in the microcontroller, run at a speed matching that of the sensor. The graphic below shows how the module will be connected and which line will be connected to what.
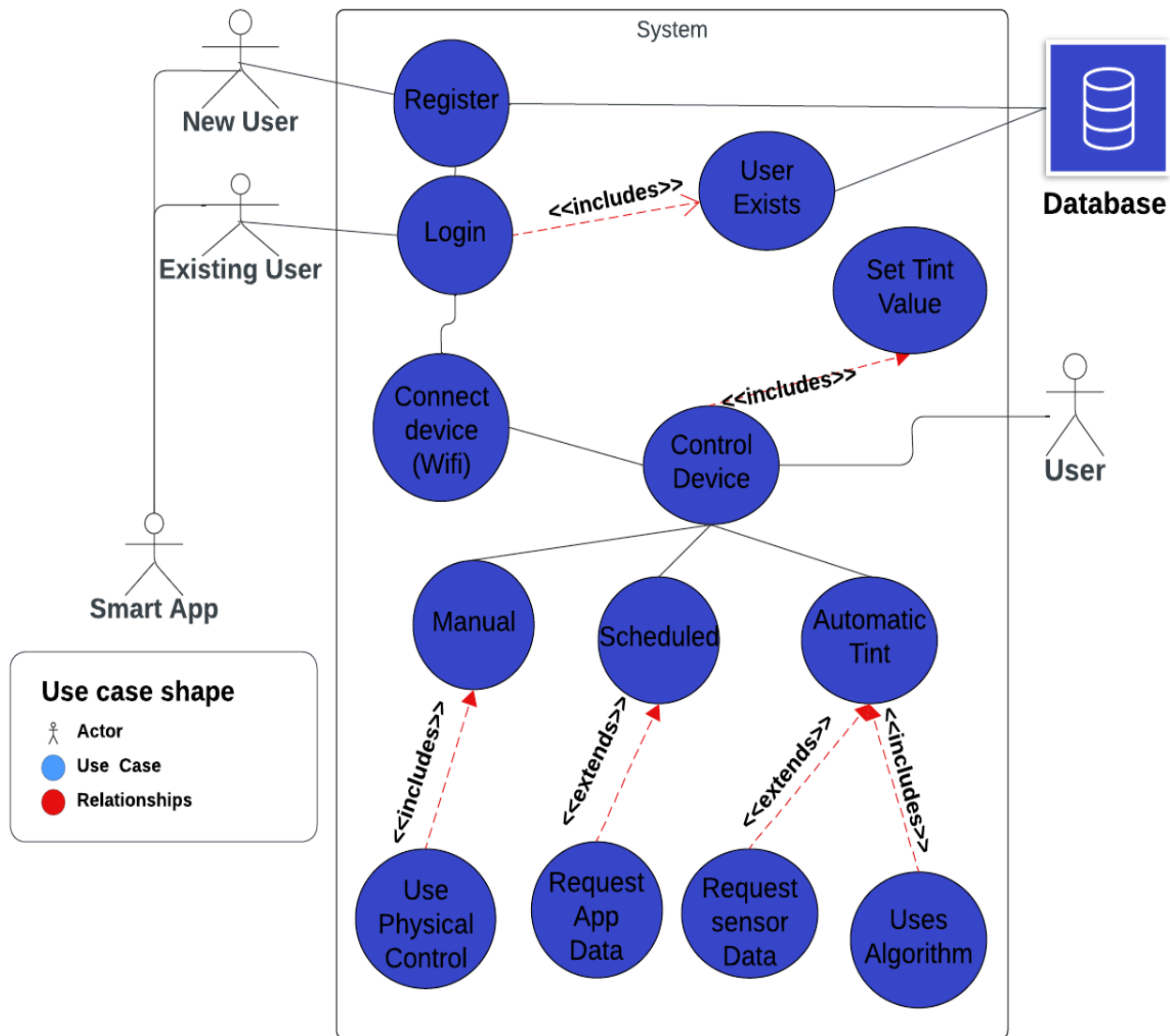
Figure 7.1 Software Use Case Diagram

The diagram above shows the different interactions between users and the Smart app connected to the control devices, which serve as the hub for the separate microcontrollers at the windows. On the left, we can see two different types of users: new and existing. New users can register, while existing users can log in. Both of these steps involve taking the information that the user has provided and verifying it in the database that we have set for the application. The system makes it easy to connect the device using WiFi and control it from there. The options for controlling are classified as manual, scheduled, and automatic mode based on the desired behavior of the window. Each will have specific pathways, including being able to control them using physical control and using the data that we can retrieve from the sensors.

The lower section of the use case diagram goes into the specifics of the use case involving the control device operations. Manual control is straightforward, just reading from the physical control. For scheduled mode, there's an inclusion for setting the time values; automatic mode involves using our algorithm from reading the temperature sensor and the light sensor and setting the best tinting values. All the different elements are interconnected, showing a comprehensive view of how the different parts of the system will interact with each other to provide the best way of managing the system functions.
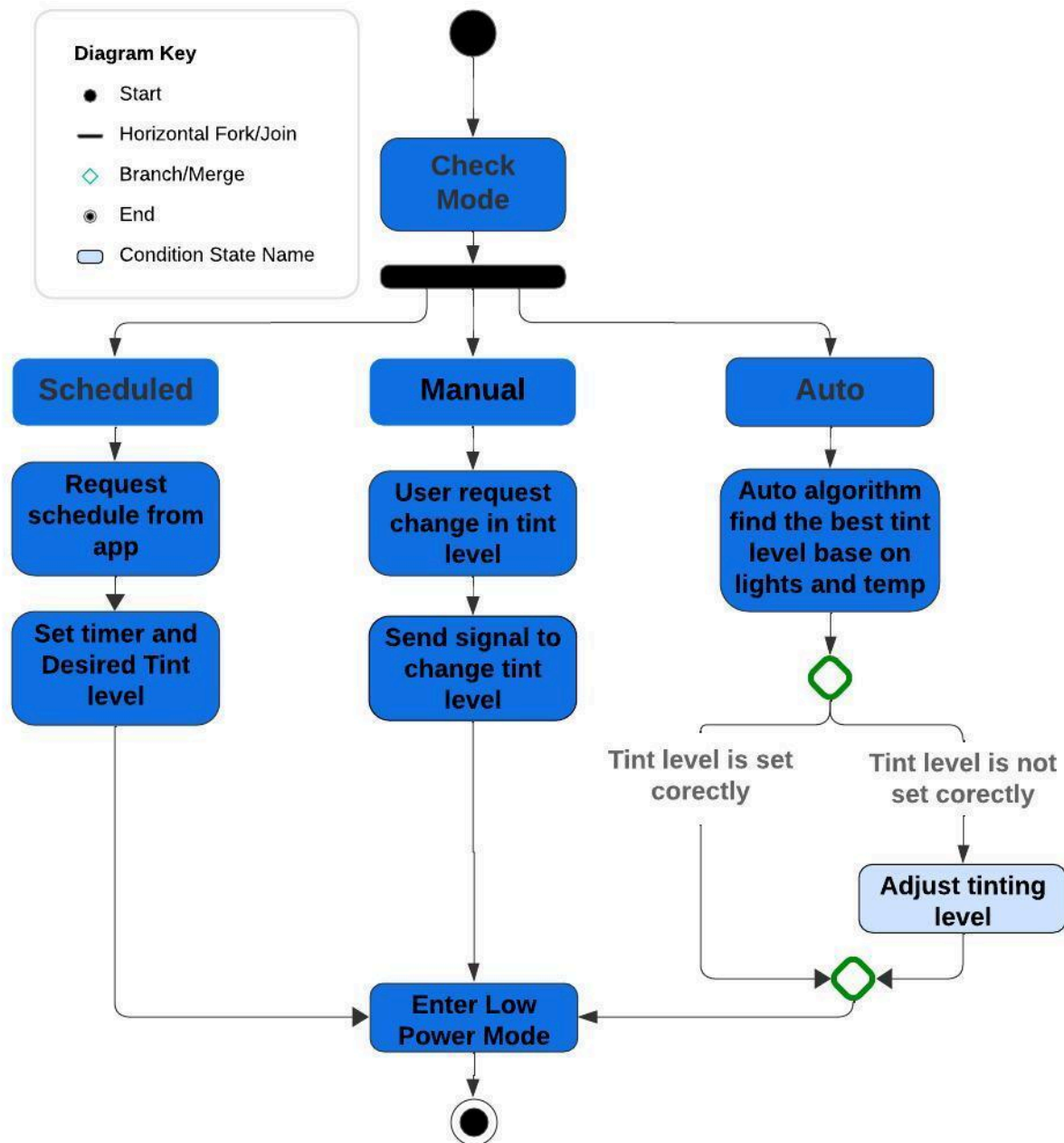
Figure 7.2 Activity Diagram

98

This diagram represents the process flow for adjusting the tint level of the tinting system. The process begins with checking the mode that leads to three different types of paths: scheduled, manual, and auto. The system in the path of the schedule will request the schedule from the app and set a time and the desired tint for that specific time. In manual mode, it responds directly to the user's request to change the tint level and sends the signal to adjust the system appropriately. The auto mode involves an automatic algorithm that determines the best tint level based on light level and temperature.

Each path converges at two decision points; whether the tint level is set correctly or not. If correct, the system will enter a form of low power mode. If it's not correct in the automatic mode, it adjusts the tinting before going into low-power mode. This diagram shows how the different modes are handled and how the different decisions are made based on the desired outcome. This diagram provides a clear overview of the function of the system and the point of interaction for the user.
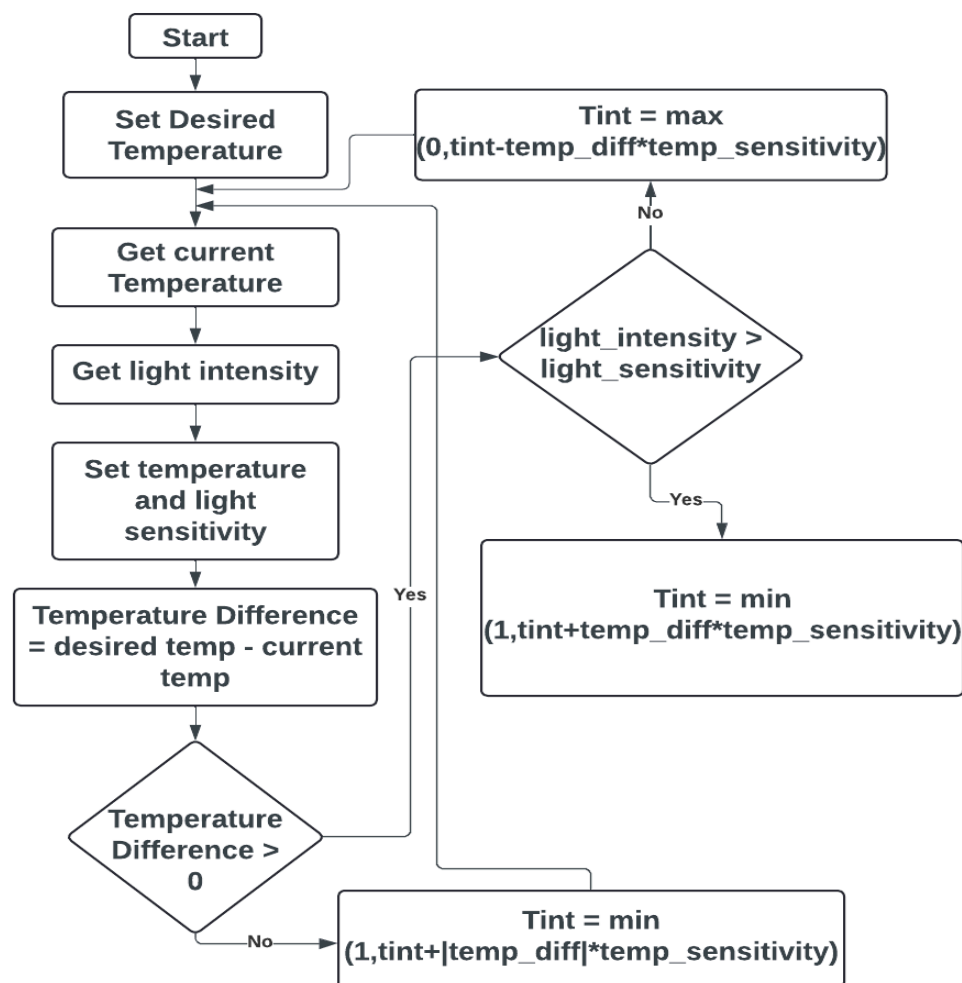
Figure 7.3 Automatic Tint Algorithm

99

This algorithm is designed to control the tinting level of the windows based on the current temperature, the sensitivity of the light, and the current tint level of the system. The algorithm begins by defining the sensitivity of the system to changes in the intensity of the light and temperature. These values can be adjusted depending on how sensitive you want the system to respond. We then calculate the differences between the desired temperature and the current temperature that we are getting from the sensors. Depending on whether the desired temperature is higher or lower, the algorithm adjusts the value accordingly.

If the desired temperature is higher than the current temperature and when we read the values from the light sensor and there is a lot of sunlight, the algorithm will decrease the tint to let more sunlight in, which would increase the temperature. If there is not a lot of sunlight, the algorithm will either maintain or decrease the tint further if possible to try to let in as much light as possible. If the desired temperature is lower than the current temperature, the algorithm will increase the level of tint to block out the sunlight and lower the temperature. This algorithm will basically use all the data that we are receiving from the sensors and put them all together to make a self-adjusting system.
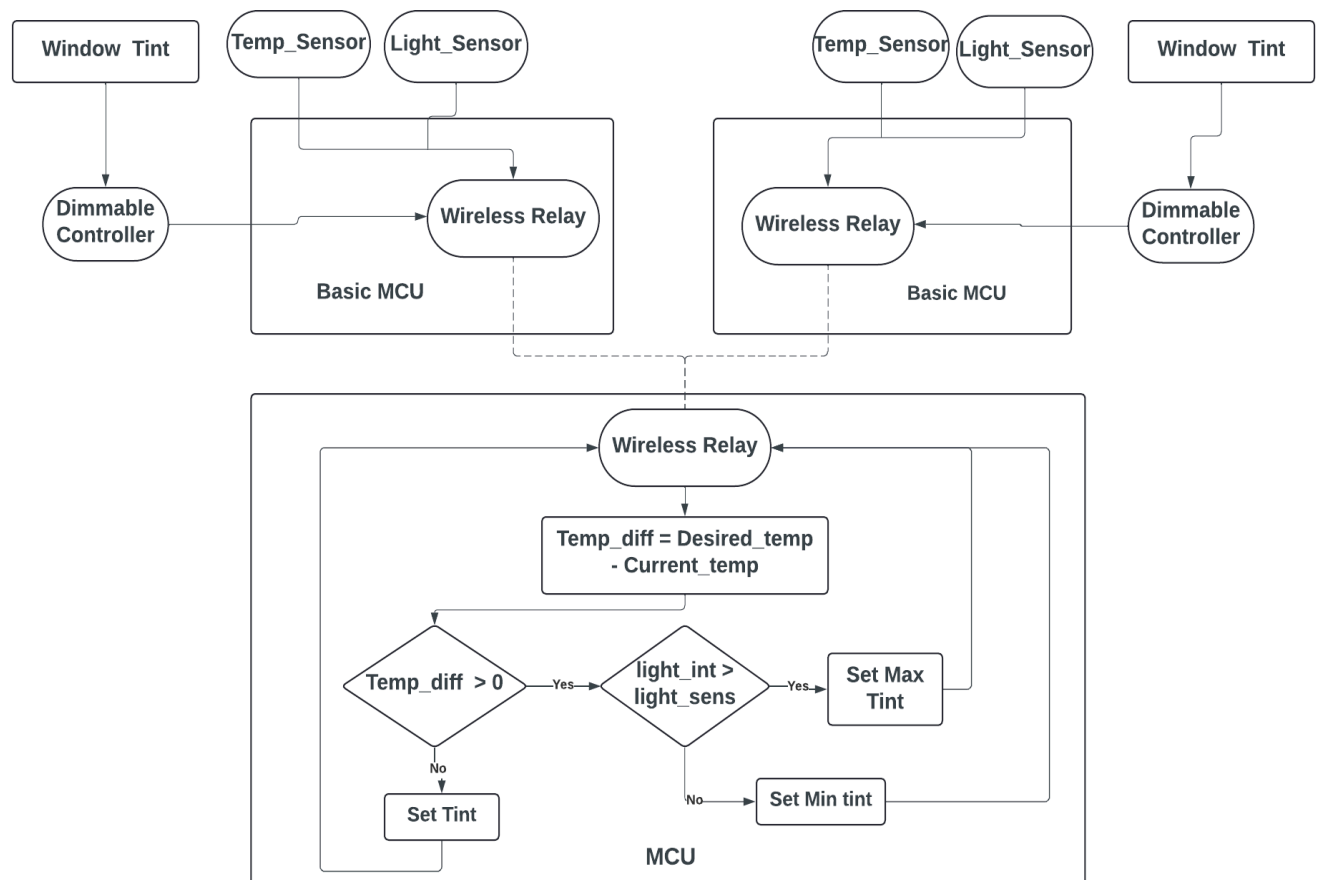
Figure 7.4 Sensors and Tint Control

First, we start by taking in the values from the light and temperature sensors to give us the amount of light that is present and the current temperature to compare to the desired temperature. These values are transmitted wirelessly and sent to the main board for the algorithm to determine

the best level of tint that will work for the current light and temperature levels. This process will allow the two separate windows to sync together at the same time if needed or if a mode requires it. This system is set up in this way to allow all the different modes to serve their functions without having to change the hardware.

After sending the values, we can then calculate the temperature difference and see if we get a value greater than 0. Then we can set the tint value by sending it wirelessly to the window board and set the value there using the dimmable controller. This process is similar if the temperature value ends up being less than 0. Then we can also use the algorithm which will give us a different value based on the intensity of the light and the temperature, and that will be sent wirelessly to change in the dimmable controller.

7.2 KnightTint Mobile Application Design

7.2.1 Login/Signup

The login page offers users the option to sign in using their existing account credentials. Illustrated below, it comprises several key components: a username field for inputting a unique identifier, and a password field for authentication. Both username and password details are securely stored in the database. Additionally, a "forgot password" link is provided, allowing users to reset their password if forgotten. Clicking this link directs users to a password reset page, where they can verify their identity via security questions and create a new password. Upon entering valid credentials, the focus shifts to the sign-in button, initiating the authentication process upon clicking. Finally, for new users without an account, a signup link is available, redirecting them to the registration page where they can create an account by providing the necessary details such as username and password.
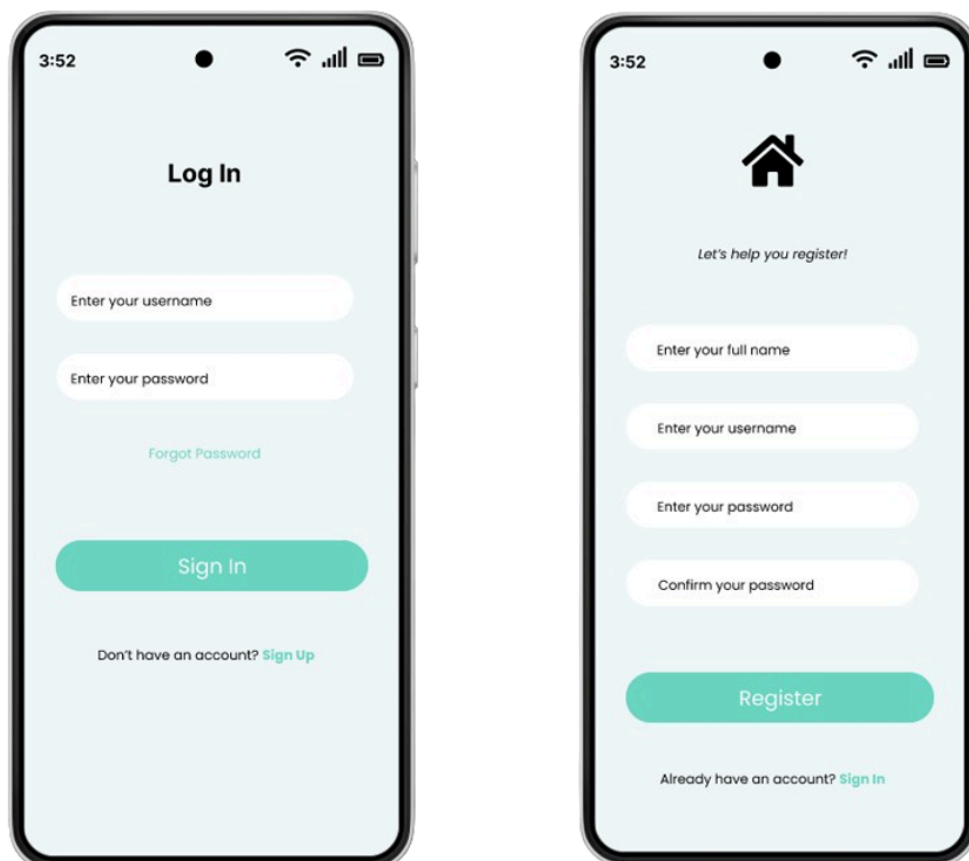
Figure 7.5 Log In   **Figure 7.6** Sign Up

The signup page serves as a gateway for users to establish a new account, catering to those who lack one or encounter difficulties with password resets or username retrieval. Its design encompasses several key components crucial for a seamless user experience. Firstly, there is a designated field prompting users to input their full name, thereby adding a personal touch to their account. Much like the login page, distinct fields are provided for the entry of both a username and password. Additionally, a confirm password feature acts as a safeguard against input errors, requiring users to re-enter their password to ensure consistency with the initial input. Upon completing all necessary fields, users are prompted to click the register button to formalize the account creation process. Furthermore, a convenient "sign in" link is readily available for users with existing accounts, seamlessly redirecting them back to the login page with a single click.

7.2.2 Landing Page

The landing page is a doorway to our mobile app for users to access all the functionality of the app. It has a nice intuitive interface designed to guide users through three different scenarios. Here's a description of the landing page layout and the three scenarios:

Landing Page Layout

The landing page features a nice design with a top navigation bar, a middle navigation bar, and a bottom navigation bar. The top navigation bar includes a dropdown menu with options like "Add a Home" and a "+" sign for adding new devices or features. The middle navigation bar is positioned below the top navigation bar, it features buttons for "Add Devices" and "Favorites." These buttons allow users to easily add new devices to their smart home network and access their favorite settings or devices with a single tap. The bottom navigation bar consists of icons for "Home," "Notifications," "Settings," and a user profile icon ("Me"). These icons provide quick access to the home screen, notifications, settings, and user profile, respectively.
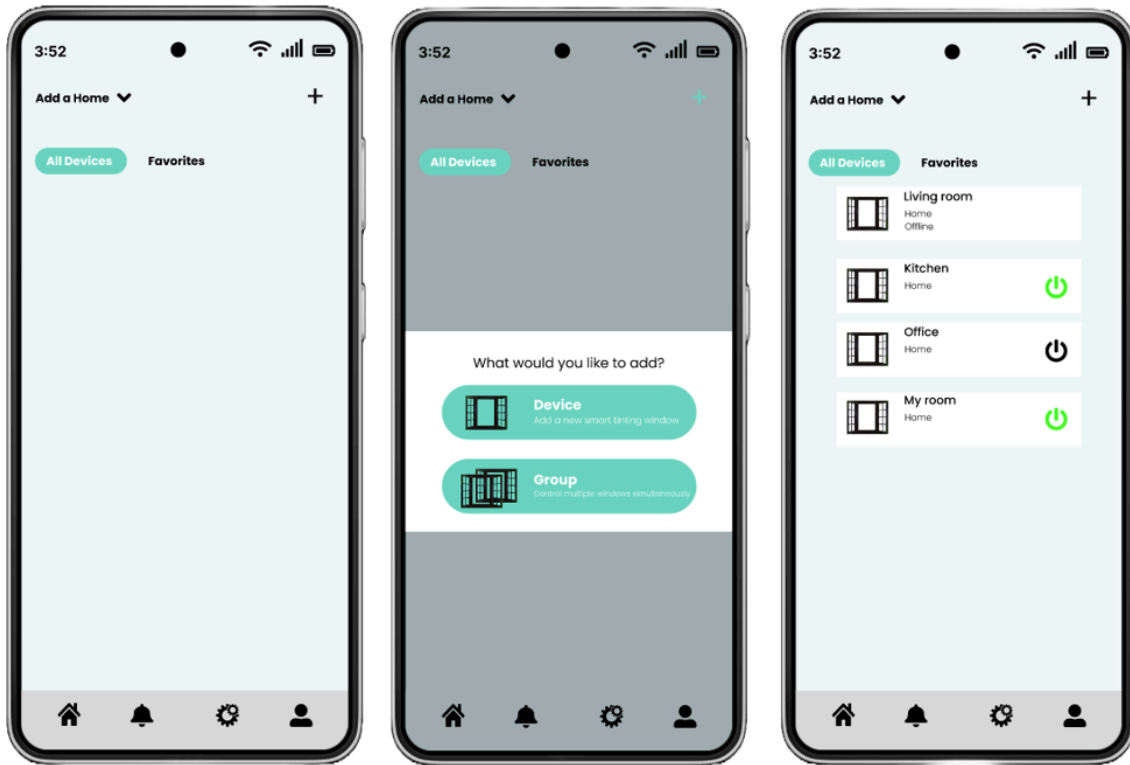
Figure 7.7 Landing Page Layout

The user navigates to the landing page and clicks on the "Favorites" button in the middle navigation bar. This action displays a list of the user's favorite devices or settings, allowing them to quickly access and control their preferred devices without navigating through multiple screens. The user can customize their favorites list by adding or removing devices based on their preferences, ensuring quick and convenient access to frequently used features.

The user clicks on the "Settings" icon in the bottom navigation bar to access the settings menu. This opens a settings page where the user can configure various aspects of the Knighttint system, such as network settings, user preferences, notification preferences, schedule, and device settings. The user can customize settings to suit their specific needs and preferences, ensuring optimal functionality and personalization of their experience.

When the user clicks on a specific window device, it directs them to a dedicated page that provides detailed information and controls for that particular device. As it is shown below, The top navigation bar displays the name of the selected window device, allowing the user to easily identify which device they are currently viewing and controlling. Below the top navigation bar, the page displays the current outside temperature. This information provides context for the user to make informed decisions about adjusting the tint levels of the window based on external weather conditions. Finally, a wheel button is provided on the page, allowing the user to adjust the tint levels of the window. This interface element allows the user to control the transparency or opacity of the window tint to regulate the amount of sunlight and heat entering the room.
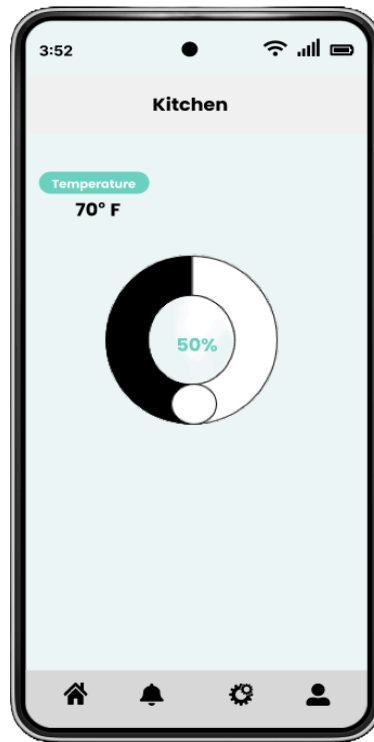
Figure 7.8 Dimmable Screen

Backend

The Knighttint backend comprises a cloud server and an accessible programming interface, enabling users to access all device features and data via the frontend. Utilizing Amazon's free tier is sufficient for efficient operations, although it may not accommodate a large number of devices. We opted for a cross-platform approach in app development to ensure availability across multiple platforms, leveraging MongoDB for the database and Express.js for the API, as shown in the figure below.
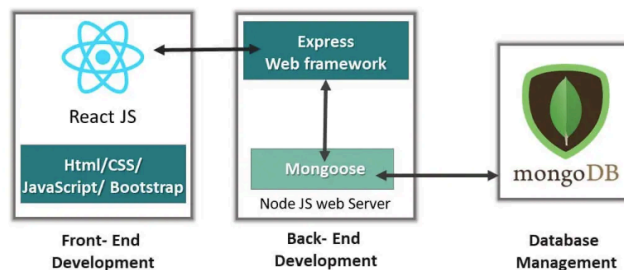
Figure 7.9 MERN Stack

8. Prototype Construction / System Fabrication

8.1 PCB Layout

Our PCB was designed in Altium Designer. Its dimensions are 6.5" by 4". PCB traces were routed through vias between the top and bottom layers to facilitate optimal routing. The top and bottom layers have ground pours. The bypass capacitors and pull-up resistors have been placed near their respective components in accordance with the datasheets and hardware design guidelines. The antenna has been placed farther away from the other components to RF noise. The PCB has a battery compartment that houses three 18650 Li-Ion batteries which will be able to power this device after being regulated down from 3.7V to 3.3V. In each of the corners, a 3.2mm mounting hole has been placed, through which we will mount the PCB using M3 screws.

Below are two images. The first image is the 2D PCB layout view on the top layer. Here, the footprints of each component, the top ground pour, as well as the routing of traces can be seen. In the second image, the 3D model of the PCB can be seen, which gives a more practical view of the PCB physical realization. Note that the DMX cable connector does not have a 3D model provided by the manufacturer, so only the footprint is present in both views.

Multiple changes to this PCB are being made. The power traces are going to be widened to ensure proper delivery. The signal traces are being separated better to ensure a reduction in signal noise.
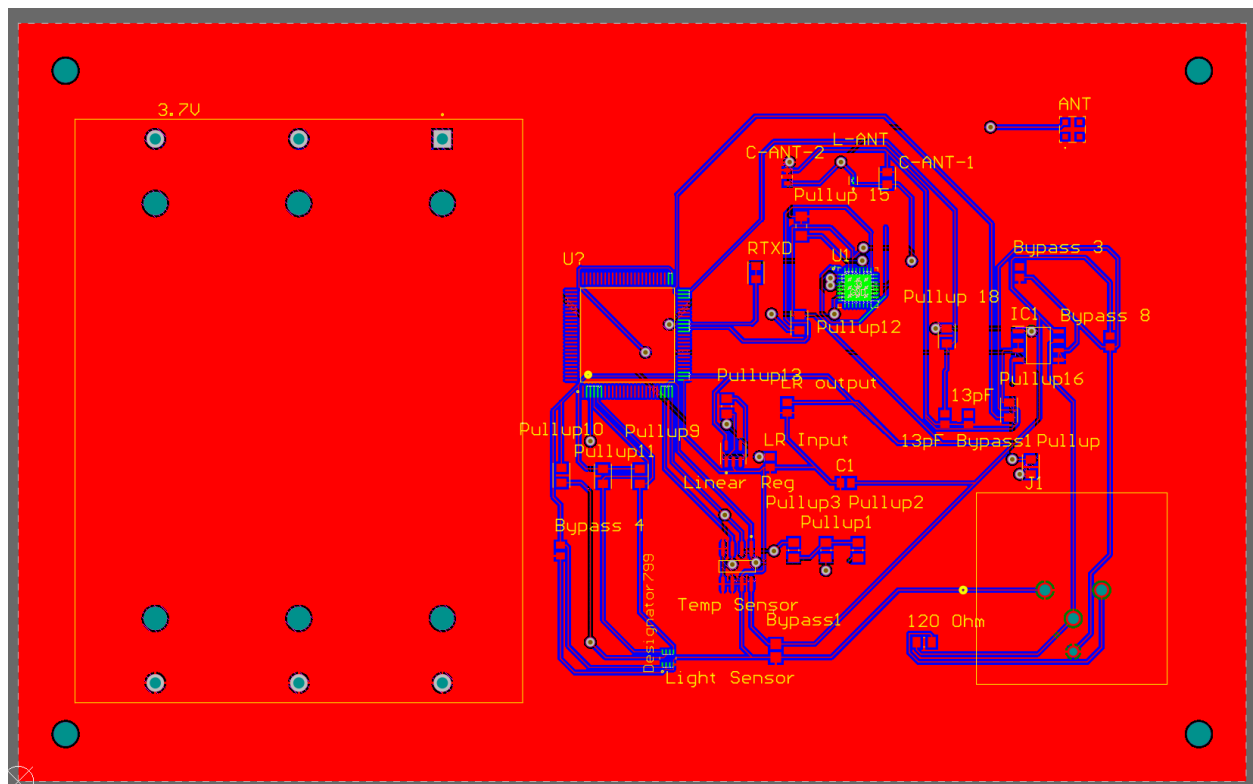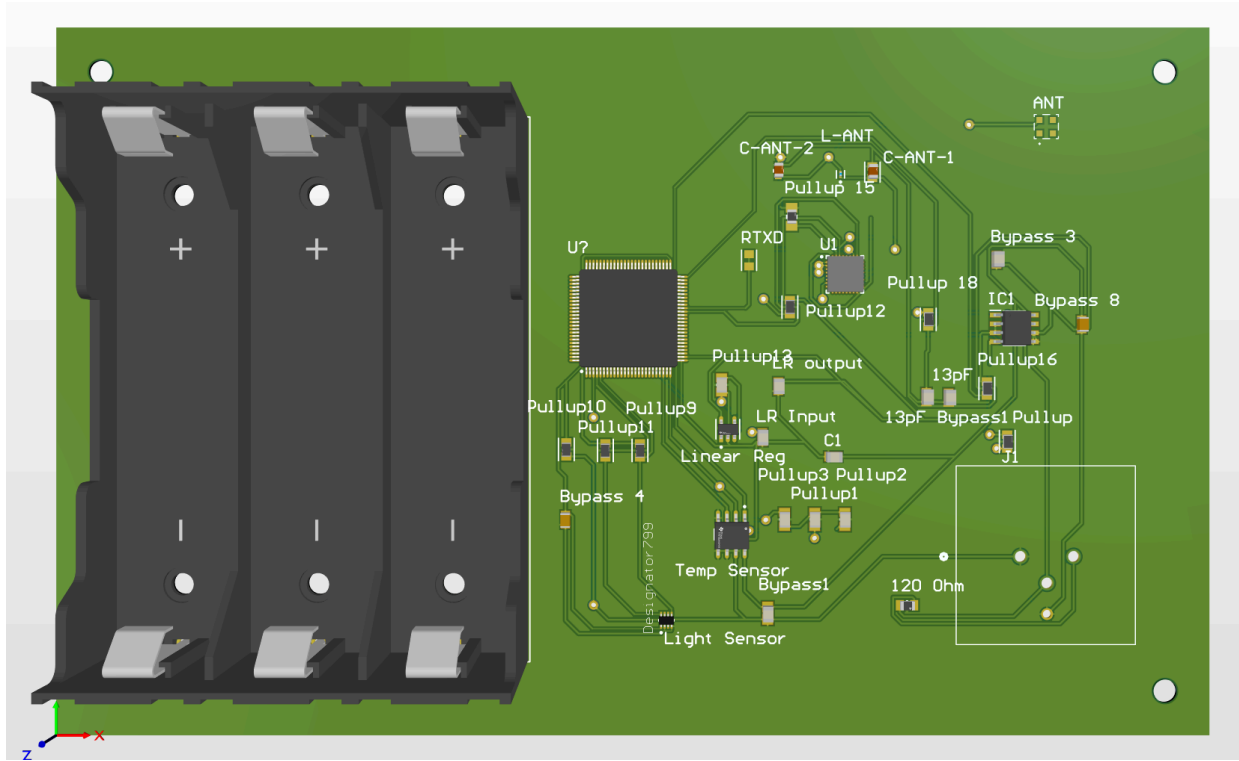
8.2 Model Housing Design

Figure 8.1 displays the latest version of our 3D CAD design for the house that will contain the window system. The design is based on a typical house format, and we have omitted some rooms to provide ample storage space for the components that will operate the tinting system. We used Onshape software to create the 3D-rendered house design.
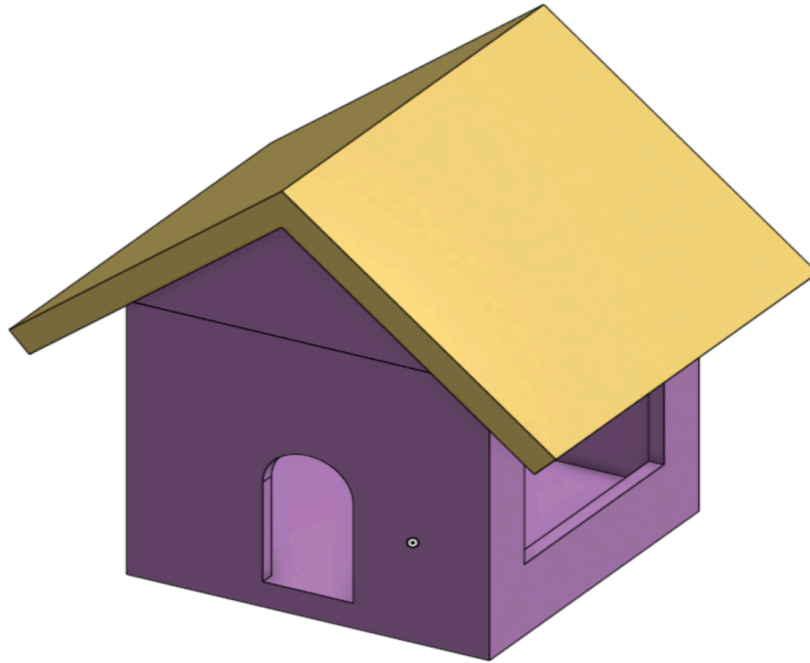
Figure 8.1 Housing Design

Our team decided to make the housing more of a square shape and have the windows be on opposite sides of each other to allow us to better demonstrate the automatic ability of the house. For example, we can have a light that shines on one side of the house and have it show that it tints that side of the house. This will allow us to also store most of the PCB that we will have inside of the house and not have to wire it in from outside. We decided to go with foam for the outside section of the house because it's a lot easier to mold and make it into the specific shape that we want. In terms of dimensions, the main constraint that we had to take into account is the size of the window, so we made it 24" x 24" x 20" because most of the sellers that we encountered were selling the window in square foot pieces. This size also allows us to keep the project at a reasonable size so that we are not having to spend a lot of time trying to actually build the house rather than the hardware we're going to put in it.
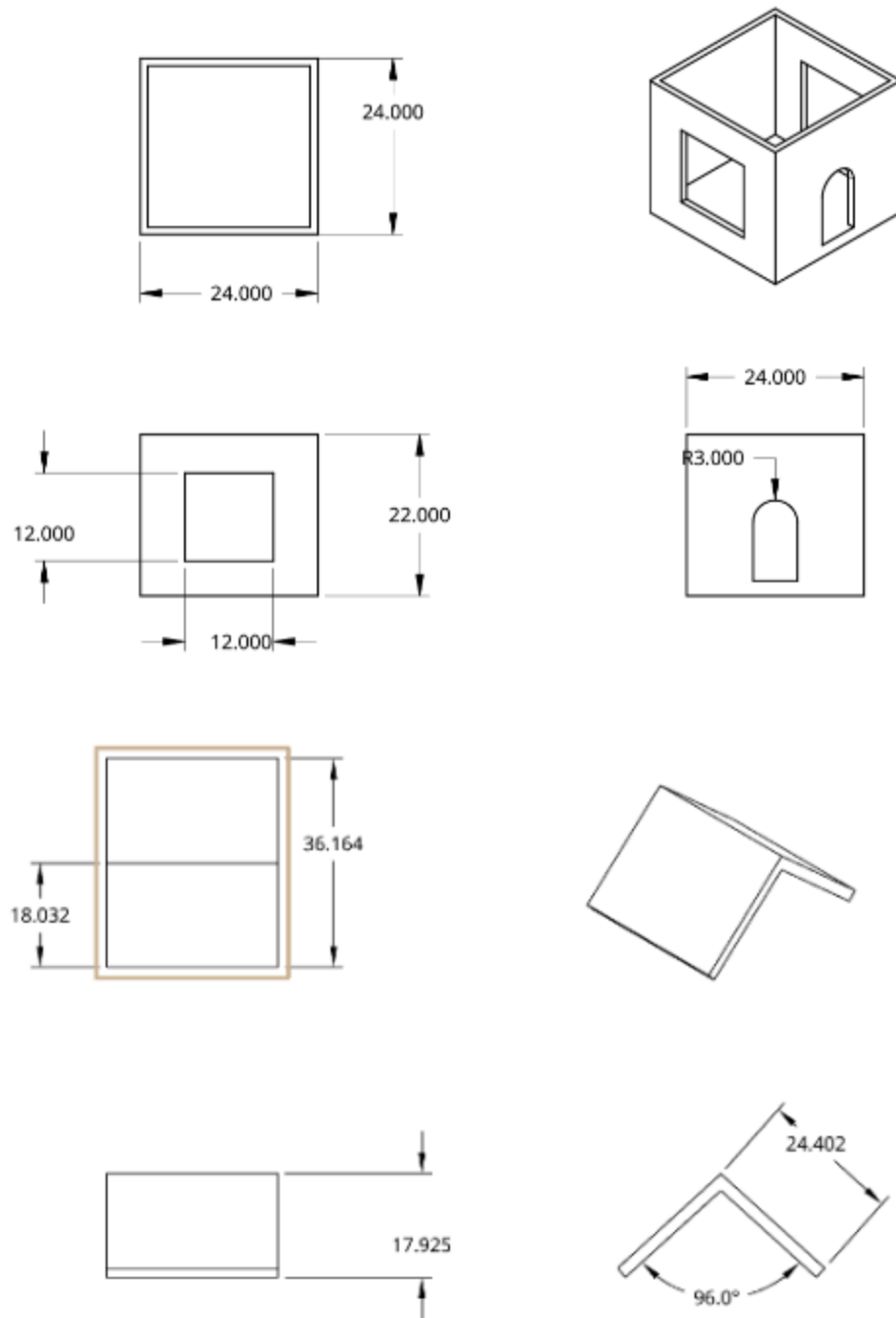
Figure 8.2 Housing Drawings

Above is a drawing that we were able to make for the body and the roof design of our house. Included are the dimensions of the main housing part that we plan on implementing. The main body on the left, we plan on building using foam and cutting it into shape. This is due to some of

the insulation effects of foam and also the ability to shape it into the different shapes that we need. This will also be useful when we start calculating the effectiveness of some of our work and trying to calculate some of the energy savings that we would get using our system. For the roof, which is the picture on the right, we plan on using wood to try to benefit from some of the insulation benefits and also add character to the housing unit.

9. System Testing

9.1 Hardware Testing

9.1.1 MSP430FR6989 Development Kit

To test that the MSP430FR6989 development boards purchased were working correctly, they were first plugged into power through their USB ports. The RGB LED powering on verifies that the device is receiving power. After this, GPIO pin voltages could be verified using a digital multimeter. The reset button functionality could be verified by pushing it.

9.1.2 CC3100 EMUBOOST

To test the CC3100 EMUBOOST we will start by connecting the CC3100 BoosterPack to the development board and ensure that the power supply is stable and all necessary connections are secure. Then we can Verify that the CC3100 can successfully connect to Wi-Fi networks and test data transmission reliability. After that we can proceed to measure power usage under different scenarios to optimize energy efficiency since that is important for our project case.

9.1.3 TMP1075DR

To test the temperature sensor we purpose an evaluation module which allowed us to test different modes and features present in the sensors. It also provides simple codes that can be altered to allow the sensor to run in different modes including having different sample time and seeing how that changes the reading that we get from the sensor. The software also provides us with the different register values needed to configure the sensor in the mode that we want for the project.
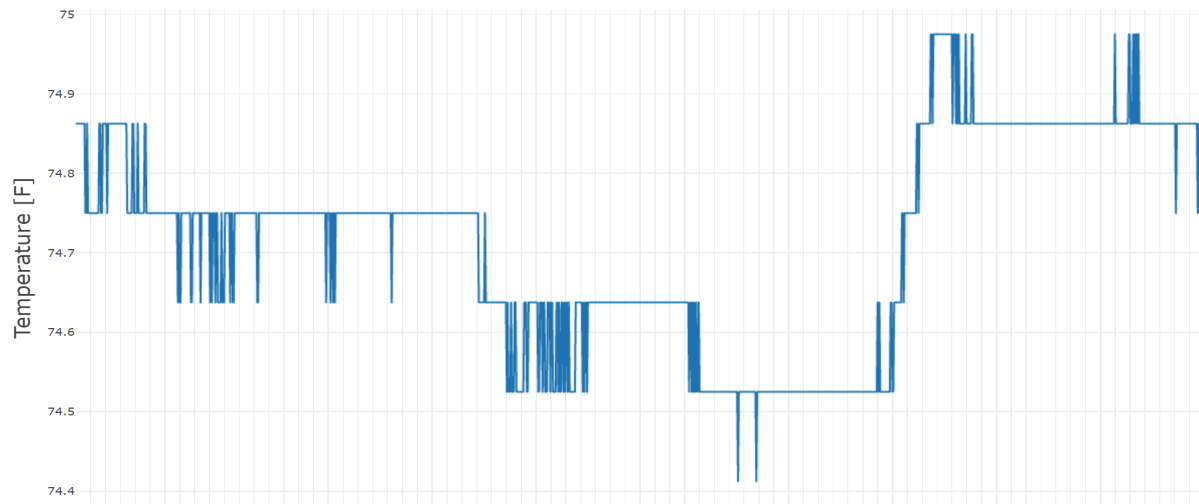
Figure 9.1.1 Temperature Sensor

9.1.4 OPT4001

To ensure optimal functionality of the OPT4001 light sensor, a software package is provided post-purchase for evaluation purposes. Upon installation, the sensor is connected to the computer via a USB port. A flashlight serves as input to the light sensor. As the flashlight is brought closer to the sensor, the lux readings increase. This experiment confirms the sensor's ability to accurately detect ambient light, making it suitable for project applications.
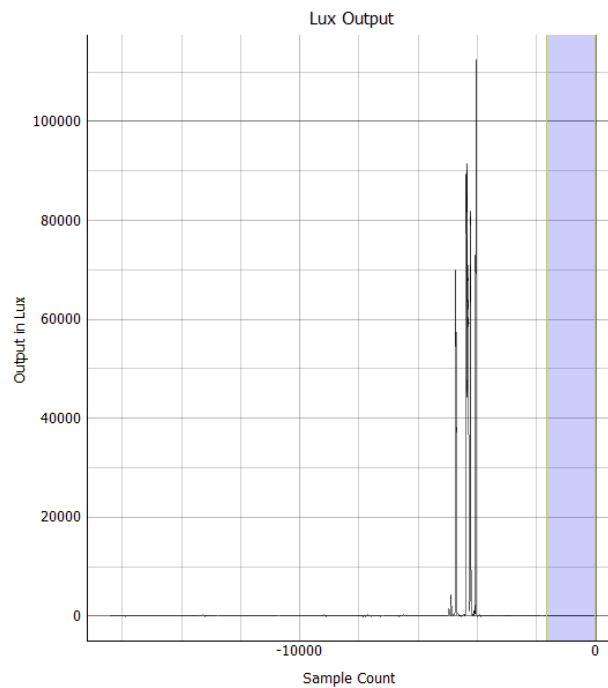

Figure 9.1.4 Light Sensor

110

9.1.5 ESP8226EX and Antenna

Testing of the ESP8226EX Wi-Fi module will occur after the window PCB has been fabricated. We will send sensor data from the ESP8226EX data over Wi-Fi to a PC and see if the transmitted data matches the environmental behavior. To achieve this, we will set up a controlled test environment where the ESP8226EX's performance can be evaluated under various conditions. This will include verifying the stability and accuracy of the Wi-Fi transmission by comparing the sensor outputs on the PC with the actual conditions recorded during the tests. Such thorough testing is crucial to ensure that the module communicates effectively and consistently, providing reliable data integration for real-time monitoring and decision-making processes in our system.

9.1.6 Batteries and Voltage Regulator

To ensure that our window boards are supplied with the correct voltage above 3.3V, we will test the voltage of our 18650 batteries with a voltage meter. This will be done for each of the batteries, as well as at the output from our battery compartment on each board. We will also use the voltage meter to test the output from the linear regulator and make sure it is 3.3V.

9.1.7 Filmbase PDLC Film

The first way of testing our PDLC will be with the transformer and ON-OFF switch provided by the manufacturer and attached to the purchased product. This will ensure that the film is not faulty and responds to voltage.

The second way of testing our PDLC film will be utilizing the Tektronix AFG3022B arbitrary function generator found in the UCF Senior Design laboratory. Although these function generators will not be able to reach the 60VAC required for the maximum translucent state, we will be able to see how the film dims at the lower voltage levels. This model outputs voltage in a range from 0-20 Vpp. We will be able to test the film's behavior with a square wave and sine wave input to see if either is more ideal.

The third round of testing for the PDLC film will occur once our dimmable controller has been tested and verified. We will output a range of voltages from 0 to 60 VAC. We will be able to see the dimming behavior of the tint, and we can use the light sensor evaluation board to measure the light transmissivity numerical value at each voltage level.

Below are the results from the first round of PDLC film testing.
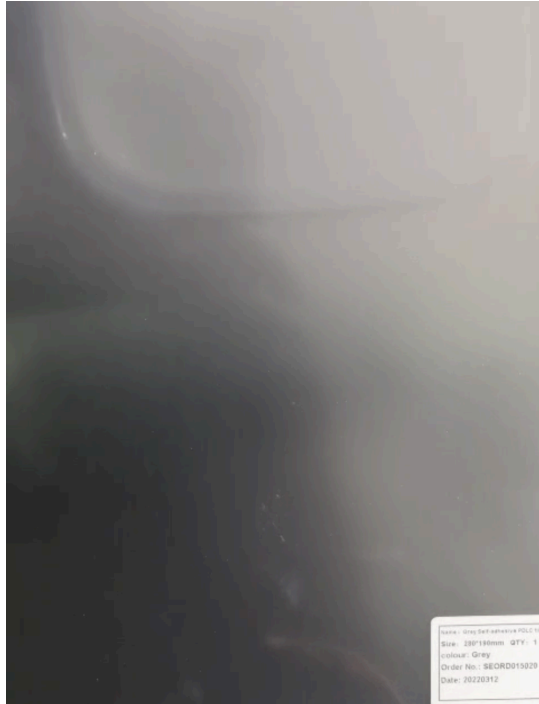
OFF   **ON**

9.2 Software Testing

9.2.1 API

The backend of the mobile app will be hosted entirely on cloud servers to give access to the Application programming interface (API) for frontend clients and automated devices to communicate with.

To ensure a modern API that effectively integrates with client-side applications, we will design it according to the principles of Representational State Transfer (REST) using JavaScript, JavaScript Object Notation (JSON) format, and Node JS for the backend service. We begin by installing Node JS which is a free, open-source server environment, cross-platform JavaScript to create the KnightTint mobile app by managing the server-side aspects of React. To start we use the npx create react app command followed by "knight-tint-app" to create the mobile app on the command line as shown in the figures below. By selecting Visual Studio Code as the primary code editor for our project, we set up the development environment for React by configuring it with extensions and settings conducive to React development, such as ES7+ React/Redux/React-Native snippets, VSCode React Refactor, and Simple React Snippets.

First, ES7+ React/Redux/React-Native snippets provides many shortcut prefixes to boost development and help developers create code snippets and syntax for React, Redux, GraphQL, and React Native. VSCode React Refactor is used to refactor code quickly and extract pieces of

JSX code into new classes, components, and so on. Last, Simple React Snippets is an extension that can easily add several React code snippets by typing a few letters.

In addition to the extensions and settings, here are some dependencies that we use to create the mobile app:

1. **bcrypt**: Bcrypt is used for hashing and securely storing user passwords to ensure authentication and security.

2. **cors**: CORS (Cross-Origin Resource Sharing) is important for allowing the frontend to run on a different domain or port, to make requests to the backend from the React app's domain.

3. **dotenv**: Dotenv allows the use of environment variables in Node.js applications. It is useful for storing sensitive information such as database credentials or API keys without hardcoding them.

4. **express**: Express is a minimalist web framework for Node.js that can be used to build the backend server for defining routes, handling requests, and managing middleware with Express.

5. **Jsonwebtoken**: JSON Web Tokens (JWT) is used for authentication. When a user logs in, it can generate a JWT that contains information about the user and send it to the client. The client can then include this token in subsequent requests to authenticate itself.

6. **mongoose**: Mongoose is an Object Data Modeling (ODM) library for MongoDB and Node.js. It provides a straightforward way to work with MongoDB through schema-based models.

7. **nodemon**: Nodemon is a tool that helps develop Node.js-based applications by automatically restarting the node application when file changes in the directory are detected.

8. **axios**: Axios is a promise-based HTTP client for the browser and Node.js. It can be used to make HTTP requests from the React frontend to the Express backend, for example, to fetch data from the server or send data to the server.

9. **React-router-dom**: React-router-dom is used to handle routing within the React app, allowing users to navigate between different pages.

10. **bootstrap**: Bootstrap is a popular CSS framework that provides pre-styled components and a responsive grid system. It is used to quickly style the React components and ensure that the app looks good on mobile devices.

Figure 9.2.1 Create KnightTint app

Figure 9.2.2 Dependencies

API User Models

Our initial step involves crafting the API user model and delineating the endpoints that enable interaction with our system, as depicted in the illustration below. The user data model encapsulates pertinent personal information pertaining to registered users, encompassing crucial details such as email addresses, names, and passwords.

To initiate user creation, we employ an HTTP POST request directed to the specified endpoint. The request payload includes the user's email address and password, which serve as the authentication credentials. Subsequently, the password undergoes encryption on the server-side, employing bcrypt for robust security in storage.

Upon successful authentication, an API access token is furnished, streamlining subsequent requests. Lastly, the login process entails sending an HTTP POST request to the designated endpoint, with the request body containing the user's email address and password. Upon successful validation, an API access token is issued, empowering users to execute future requests securely.
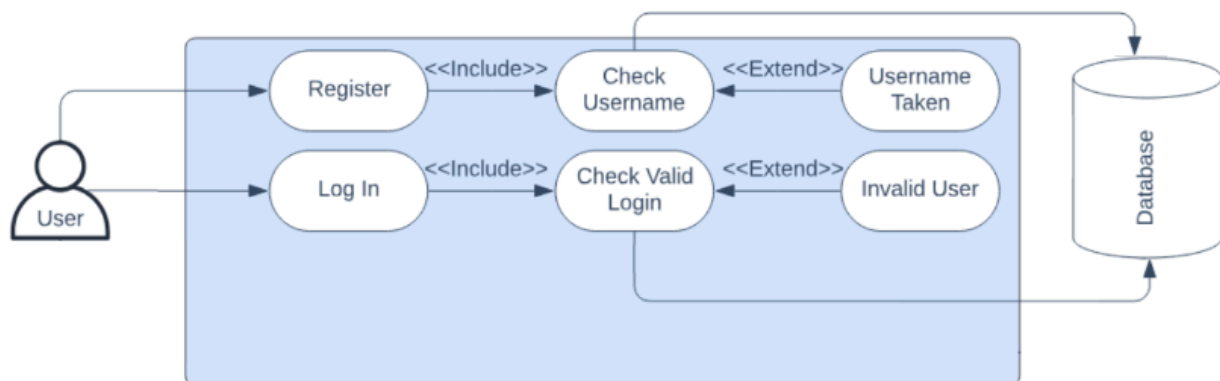

Figure 9.2.3 API

9.2.2 Database

The database of our project is on MongoDb which makes it important to install MongoDB Compass as the GUI for MongoDB for querying, optimizing, and analyzing data.

User Schema Models

The User model will contain user information and global settings. Specifically, it will primarily include a unique ID for referencing the user object in other models, the user's email address, and the user's password. Within the database, we will be indexing the User model by ID as well as email address. User IDs will be resolved within references of other models, while email addresses will be utilized for logging in.
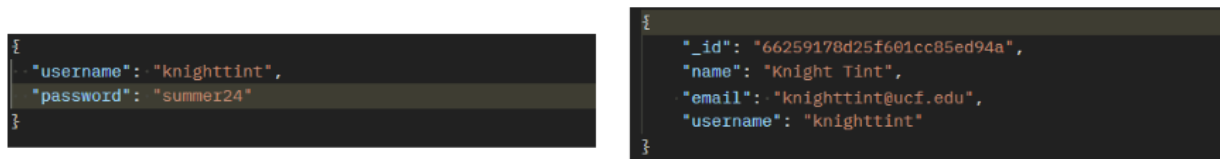
Below are some testing outputs following the installation and implementation of the backend, which include both the API and the database using Podman and MongoDB Compass.

Figure 9.2.2.1 Signup - Podman

Figure 9.2.2.2 Login - Podman

Figure 9.2.23 User data - MongoDB Compass

115

10. Administrative Content

10.1 Project Budgeting and Financing

Budgeting and financing are crucial for the success of the smart tinting window project. Acquiring the necessary components is estimated to cost around $500.00 USD, with each of the four members contributing $125.00 USD. This even distribution ensures a fair financial commitment, covering the basic expenses for the project's success. Several other costs were accrued during our time in SD1. Evaluation boards for our 2 sensors ended up costing $140. We ordered extras for most of our devices to ensure that we would get working PCB's. This cost us an additional $160.

Item	Total Price (USD)	Quantity
PCB	$140.00	3
Smart Glass	$240.00	2
Microcontroller	$30.00	1
Connectivity Devices (Wi-Fi/Bluetooth)	$20.00	3
Demo House	$50.00	1
Light Sensor	$10.00	2
Temperature Sensor	$10.00	2
Total	**$500.00**	

10.2 Initial Project Milestones for Each Semester

Semester I (Senior Design)

Week #	Dates	Milestone Description
1-3	1/8/2024 - 1/29/2024	Complete and Submit the divided and conquer assignment and start working on the research portion of the paper.
3-6	1/29/2024 - 2/19/2024	Complete and Submit a 60-page draft and meet with advisors to further discuss ideas.

6-9	2/19/2024 - 3/4/2024	Start ordering parts begin elementary testing and continue to work on the 120-page paper
9-10	3/4/2024 - 3/18/2024	Continue to work on the 120-page paper
10-13	3/25/2024 - 4/15/2024	Begin working on an initial PCB
13-16	4/15/2024 - 4/29/2024	Start working on and finish initial testing prototype

Semester II (Senior Design)

Week #	Dates	Milestone Description
1	5/13/2024 - 5/20/2024	Append to the 120-page document thing we've learned from the prototype.
2	5/20/2024 - 5/27/2024	Individual System design (APP, MCU, PCB)
3	5/27/2024 - 6/3/2024	Continue working on PCD MCU and APP (update 120-page paper)
4-9	6/3/2024 - 7/8/2024	System integration (Make sure everything is working together and connected)
9-10	7/8/2024 - 7/22/2024	Practice project demo (Go through a knock demonstration in front of our committee members)
11	7/22/2024 - 7/29/2024	Finalize project demo and documentation
12	TBD	Final Presentation

10.3 Work Distributions

To enhance the coherence of our team's project workflow, we decided to define specific roles for each member involved in this project. These roles are designed to cluster related tasks within the project, preventing potential role ambiguity among team members. Addressing role ambiguity is crucial to avoid hindering project progress, maintaining efficiency, and ensuring a balanced distribution of workload among team members. It's important to note that the roles outlined here are flexible, and if a team member wishes to contribute to a project task outside the scope of their designated role, they are encouraged to do so with the majority agreement of the remaining team members. The roles utilized in this project, along with brief descriptions of each role and the team members assigned to them, are as follows:

- **Project Manager -** *Structured team meetings, established goals and objectives, and assigned different roles to team members to ensure the timely completion of all necessary tasks for the project's smooth progression. The project manager was designated as the primary representative for direct interaction with the instructors of the Senior Design course. They consolidated questions and issues raised within the team related to the project, facilitating efficient and effective resolution by instructors, other team members, or additional mentor figures.*
  - *Assigned to:* Oren Muszkal

- **Software Engineer -** *This role is designated for project team members with experience in programming software, specifically focusing on mobile app development. In undertaking this role, individuals are expected to engage in a diverse array of tasks that span both frontend and backend but also entail the crucial aspect of effective communication with hardware and embedded engineers. It is imperative for the incumbent to conduct thorough research and meticulously choose the most optimal software solutions, ensuring seamless integration and development aspects of crafting innovative and user-friendly mobile applications.*
  - *Assigned to:* Luckner Ablard

- **Embedded Software Engineer -** *Responsible for the microcontroller programming that manages the tinting system including the code to collect and process the data from the sensors, control the film, and communicate with other components. This person is also responsible for working with the hardware and software engineer to integrate the temperature and light sensor into the system. The Embedded Software Engineer is also responsible for the automation logic that takes in all the information from the sensor and adjusts the tinting level accordingly. As part of this role, this person is also responsible for selecting the appropriate microcontroller platform for the tinting system. They work on the integration of the microcontroller with other components and systems that form the main tinting system.*
  - *Assigned to:* Emmanuel Levasseur

- **Hardware Engineer -** *This role is assigned to all team members who are involved in the physical aspect of the project. Responsible for PCB design, sensors, and power management. Members assigned this role have experience and a strong background in transistors, diodes, PCBs, and other integrated circuitry.*
  - *Assigned to:* Oren Muszkal, Stephen Polner

- **Mechanical Engineer -** *This role is specifically for the building of the prototyping of our devices. This role has been tasked with the construction and design of all non-electrical components in our product. The member assigned this role must have prior knowledge in woodworking and constructing models.*
  - *Assigned to:* Emmanuel Levasseur

- **Project Documentation -** *This role involves active participation in meetings, capturing essential details such as discussions, decisions, and action items. The responsibility of*

this grows as we go further in this project when we start making decisions in which direction to go with different aspects of the project.
- ○ *Assigned to: Emmanuel Levasseur*

10.4 Project Management Tools

In this section, we will discuss the basic technologies that are used to keep the team well-organized, meet deadline management, and deliver quality results. These technologies help with coordination and communication to facilitate planning, executing, and monitoring tasks.

Discord

Discord is a free communication platform that allows people to share voice, video, files, and text via the Internet from any phone, computer, or device that supports it. It has a user-friendly interface where it's easy to navigate and utilize its features. For instance, new users can quickly join channels, and voice chats, and share screens and files without extensive tutorials or guidance. Discord is a reliable technology platform that ensures real-time communication, low-latency voice, and video communication and provides access to set up meetings for different topics with members from different time zones. It is a platform used by everyone both recreationally and professionally. Here at UCF both instructors and students communicate. In addition, the Discord mobile app is compatible where users can switch from desktops or laptops to mobile during a meeting to participate in voice calls, respond to messages, and share files from their smartphones or tablets. Using Discord as a platform of communication for our project is an excellent choice due to its flexibility, user-friendly interface, and features that allow us to collaborate and communicate effectively.

GitHub

GitHub is open-source cloud-based software that allows developers to create, store, manage, and share their code. It is a combination of Git and version control principles. Git is control system software that provides access control, bug tracking, software feature requests, task management, and continuous integration for every project. Version control that allows developers to track and manage changes to a software project's code. Our project will contain several repositories to keep everything organized. For example, a repository that contains all its source code, documentation, and related assets and another for guidelines. GitHub ensures effective collaboration and task management where developers can prioritize and assign issues to specific members for resolution. It has a pull request feature where a team member can propose changes and ask for code review and discussion before merging. In addition, a team member can create branches to work and change without affecting the main codebase. Finally, GitHub is a secure platform that helps detect common coding mistakes and security vulnerabilities during code review. Our team will use GitHub for our project because of its tremendous features to control, manage, share knowledge, document, and enhance productivity and code quality.

Altium

Altium Designer is a comprehensive, integrated electronic design automation (EDA) tool used by engineers and designers to develop printed circuit boards (PCBs) and electronic systems. It combines schematic capture, PCB design, and programmable logic (FPGA) development within

a single application, streamlining the process of bringing a concept to production. Altium allows for the management of component libraries, ensuring that designers have access to the latest parts and their specifications. It also supports version control principles, enabling teams to track and manage changes to their designs efficiently.

In projects that require coordination across multiple team members, Altium facilitates collaboration through its cloud-based platform, Altium 365, which connects PCB design to the broader electronics product design process. This environment allows for the sharing of PCB designs and manufacturing data with team members, stakeholders, and manufacturing partners, ensuring everyone is aligned and up-to-date. For instance, one project workspace might contain the PCB designs, while another could store component libraries and design guidelines, helping to keep the project organized.

Altium Designer also includes features for design rule checking, which automatically identifies potential issues in the PCB layout, such as spacing conflicts or unconnected pins, reducing the risk of errors and rework. Its integrated simulation tools allow designers to test and validate their circuits early in the design process, ensuring functionality and performance before prototyping.

With support for multi-board assembly, designers can work on complex projects involving multiple PCBs within a single project environment, viewing and managing the interconnections and spatial relationships between individual boards. This capability is critical for developing advanced electronic products that require integration of several PCBs into a final assembly.

Furthermore, Altium emphasizes security and data integrity, providing access controls and activity tracking to protect intellectual property and project data. The platform's emphasis on collaboration, design integrity, and productivity makes Altium Designer a valuable tool for our team aiming to accelerate the development of high-quality electronic products.

Google Docs

With the many different activities our group members juggled, finding software that facilitated seamless collaboration was crucial for writing our group paper. We explored numerous options, ultimately settling on Google Docs. The real-time collaboration feature was a game-changer. Instead of sequential writing, multiple members could work on the document simultaneously. This enabled efficient brainstorming sessions, collaborative writing of different sections, and smooth integration of edits.

Google Docs also proved invaluable with its version control feature. Every change made to the document was meticulously tracked, allowing us to identify who edited what and revert to previous versions if necessary. This ensured everyone was on the same page and eliminated the frustration of conflicting edits. Early in the project, we encountered some confusion regarding who was working on which sections. Thankfully, version control allowed us to retrace our steps and restore the document to a point where everyone was clear on their responsibilities. This saved us from having to rewrite or redo entire sections.

Beyond simultaneous editing, Google Docs fostered communication and feedback exchange through comments and suggestions. We implemented a peer review system where group members reviewed each other's sections, leaving comments and suggestions for improvement. This collaborative approach significantly enhanced the overall quality of the paper.

However, we did encounter a minor hurdle with page count discrepancies. The information displayed in Google Docs didn't always match our external word processing software. Despite this, the efficiency and user-friendliness of Google Docs far outweighed this minor inconvenience. The platform proved to be an essential tool, allowing us to complete our project collaboratively and effectively.

10.5 Bill of Materials

Schematics BOM

Below is a bill of materials for the components shown in the schematics section of this report. This is based on the BOM generated by Altium, with some modifications made for clarity and ease of use. This BOM includes all IC's, as well as all RLC components for KnightTint. Shown is the part number that designates the specific component, a short description to give a basic idea of what the component is, and the parameters and values that are important to each component. At the right is the minimum quantity needed total for two custom PCB boards. We have ordered more than the quantity listed for insurance against accidents and for use in any updated designs.

Part Number	Description	Details / Component Parameters	Quant.
BK-18650-PC6	Battery Holder	18650 3 Cell	2
08051A130JAT2A	Capacitor	13 pF ±5% 100V Ceramic Capacitor C0G, NP0 0805 (2012 Metric)	6
MCU08050C1200FP500	Resistor	1.05K OHM 0.5% 1/5W 0805	2
Molex 47948-0001	2.4 GHz Antenna		2
C0805F103K5RACTU	Capacitor	Ceramic, 50V, 10% +Tol, 10% -Tol, X7R, -/+15ppm/Cel TC, 0.01uF, 0805	2
Capacitor 100 nF +/-10% 50 V 0805	Capacitor	100 nF, +/- 10%, 50 V, 0805 (2012 Metric)	4
C0805C105K3PACTU	Capacitor	Multilayer, Ceramic, 25V, 10% +Tol, 10% -Tol, X5R, 15% TC, 1uF, Surface Mount, 0805	6
GQM2195C2E2R4BB12D	Capacitor	2.4pF ±0.1pF 250VDC C0G 0805 Paper T/R	2
GQM2195C2E3R0BB12D	Capacitor	3pF ±0.1pF 250VDC C0G 0805 Paper T/R	2
ST4E1240DT	RS485 Transceiver	High-speed 3.3 V / 5 V	2
AC3MAV-AU-PRE	XLR Port	PCB-Mounted	2
LQP03TN2N2B02D	Inductor	Film type RF 2.2nH ±0.1nH 600mA 0.15Ω 0201 (0603)	2
OPT4001DTSR	Light Sensor		2
TLV70033DDCR	Linear	Fixed Positive LDO Regulator, 3.3V,	2

	Regulator	0.25V Dropout, PMOS, PDSO5	
CRCW08055K00FKTA	Resistor	Metal Glaze/thick Film, 0.125W, 5000ohm, 150V, 1% +/-Tol, 100ppm/Cel, Surface Mount, 0805	8
RG2012P-123-B-T5	Resistor	Automotive Metal Thin Film Chip, 0805, 12kΩ, 0.1%, 25ppm/°C, 0.125W, 150V	2
AC0805FR-0710KL	Resistor	Metal Glaze/thick Film, 0.125W, 10000ohm, 150V, 1% +/-Tol, 100ppm/Cel, Surface Mount, 0805	14
ERJ-6ENF4990V	Resistor	SMD 499 OHM 1% 1/8W 0805	2
TMP1075DR	Temperature Sensor		2
ESP8266EX	Wi-Fi Chip		2
MSP430FR6989IPZ	Microcontroller		2

Non-Schematics BOM

Below is another BOM for the components that are not part of the PCB fabrication, including materials related to the development board, the dimmable controller, external wiring, and the model house. The quantity shown is the minimum quantity needed for the entire KnightTint system.

Item Description	Quantity
Filmbase Gray PDLC Tint	2
MSP430FR6989 Development Board	1
CC3100Boost Wi-Fi Booster Pack	1
AmazonBasics USB-A Charger	1
Micro USB Cable	1
DMX XLR Cable	2
Gauzy LCG Flex Controller	2
Controller to Tint Dry Contact Wiring	2
M3 PCB Mounting Screw	8
Wood	4
Foam	4
Screws	15

11. Conclusion

This paper goes through a comprehensive journey of the design and development process for our window tinting system, KnightTint. This project began with a lot of conceptualization on what our project should be about. As a group we individually brainstormed on different ideas and presented it to one another till we were able to unanimously find an idea that we were all interested in and provided us all with the work that we wanted to explore and more. While many ideas were discussed, the tinting system emerged as the main system that we wanted to develop. This idea related to our lives in the sunshine state of Florida and helped solve a common issue of energy costs related to indoor air conditioning, as well as built off of the smart-home movement that has been occurring over the last few decades.

We had the choice between a car system and a housing system. The car system would have implemented the tinting as an embedded part of the car and we would have had different situations like when the car started and when the car was put in park and the tinting levels would have been adjusted based on that. The main issue that arises with implementing KnightTint is that we would have to get into the car's computer system and have to learn whatever algorithm language that is being use to program the car and we would also have to use it in a car and that will involve using one of our car or get a car for this specific case which will end up putting us out of budget or damaging one of our car system. We ended up going with the housing system because it provided us with more flexibility when it came to testing by building a demo housing which will cost less and have much less risk to any of our group members.

After choosing our product, we began our research phase, where we looked into all the existing solutions available and also examined some of the technologies that we plan to use for our product. The development process was iterative, involving us researching and looking into various types of technologies and examining the advantages and disadvantages that come with using one technology over another. We started by coming up with a set of goals and requirements that we wanted our system to meet and looking at those goals we looked into all the different technologies and products that will best allow us to achieve these goals. One key focus was ensuring that the system works together seamlessly. This was achieved through extensive research and examination of datasheets to ensure that all the specifications for the parts match and are compatible with one another. We had to find a balance between technology/product availability, complexity and cost, but we believe that we made the optimal decisions for the goals of our project given that this is a self-funded prototype of our idea.

While as a group we are confident in the research that we conducted and the work that we put in to ensure that these parts all work together, there is no doubt that in our Senior Design II course, there might be some issues that arise with parts that we did not foresee. We have implemented some redundancies and backups to ensure that when these problems arise, there is room to make the necessary changes to ensure that we still have a working prototype to present at the end of this project.

One major challenge that we had to face as a team is acquiring the window tint film that we will use in the system. Most manufacturers that we approached primarily deal with construction companies, which made it hard for us to acquire any of the film. These companies typically have

order requirements that go beyond the scope of anything that we would need or use for this project, both in terms of quantity and cost. Moreover, these manufacturers often already have relationships with these construction companies, making it difficult for someone working independently or in a small team to negotiate with them. This is further complicated by the fact that construction companies usually have systems in place to handle large-scale orders and deliveries. Despite these challenges, we have navigated this landscape by contacting these manufacturers, leveraging our connections, and exploring alternative sourcing options. If we were to produce KnightTint at a more professional/commercial level, we would utilize the more expensive and exclusive technologies and products in order to make a better quality product, but for the purposes of this prototype, we are satisfied with our cost-effective product decisions and design solutions. This experience has shown us the importance of resourcefulness and persistence in the face of challenges.

After acquiring some of the components we planned to use in our project, we proceeded to start testing them to ensure they worked as expected. For the temperature sensor and light sensor, we purchased evaluation boards that allowed us to test the different sensors and make sure they were working as advertised.To test the temperature sensor's limits, we found we had to create different environments where we had control over factors like the actual temperature. We then checked the measurements to ensure we were getting values close to what was expected. The same process was repeated for the light sensor. However, measuring its accuracy proved to be much easier since the amount of light can be more easily controlled compared to the temperature in a room. This whole process allowed us to learn more about the parts we plan on using and equipped us with the ability to better troubleshoot any future issues that might come up.

As we conclude this section of our project, we look forward to the start of Senior Design II. This class will provide us with the opportunity to put theory into practice and implement some of the concepts outlined in this paper. We are fully aware that there will be some challenges that will arise, whether it be in sourcing materials or troubleshooting design issues. We look forward to these challenges as they will provide us with opportunities to learn and grow. We anticipate these experiences, confident in our ability to navigate any hurdles we may face. As we move forward, we carry with us the knowledge we have gained, the excitement for what lies ahead, and the determination to make our vision a reality.

Appendix A – References

[1] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., … Scialom, T. (2023, July 19). *Llama 2: Open Foundation and fine-tuned chat models*. arXiv.org. https://arxiv.org/abs/2307.09288

[2] *Why 32-bit mcus are not always the superior choice? comparing 8-bit, 16-bit, and 32-bit microcontrollers*. Components101. (n.d.). https://components101.com/article/difference-between-8-bit-16-bit-and-32-bit-microcontrollers

[3] 8-bit microcontrollers - MCU – mouser. (n.d.-a). https://www.mouser.com/c/semiconductors/embedded-processors-controllers/microcontrollers-mcu/8-bit-microcontrollers-mcu/

[4] Choosing the optimal low power MCU. (n.d.-b). https://eu.mouser.com/applications/low_power_choosing_mcu/

[5] GeeksforGeeks. (2023, April 15). *Advantages and disadvantages of Microcontroller*. https://www.geeksforgeeks.org/advantages-and-disadvantages-of-microcontroller/

[6] *General purpose microcontrollers (mcus)*. General Purpose Microcontrollers | NXP Semiconductors. (n.d.). https://www.nxp.com/products/processors-and-microcontrollers/arm-microcontrollers/general-purpose-mcus:GENERAL-PURPOSE-MCUS

[7] Maximintegrated. (n.d.-c). https://pdfserv.maximintegrated.com/en/an/AN2394.pdf

[8] Media, O. (n.d.). *Java for embedded systems: Advantages and disadvantages*. Embedded Computing Design. https://embeddedcomputing.com/technology/software-and-os/ides-application-programming/java-for-embedded-systems-advantages-and-disadvantages

[9] Tomlinson, Z. (2018, December 10). *What are microcontrollers and why should you care?*. Interesting Engineering. https://interestingengineering.com/innovation/what-are-microcontrollers-and-why-should-you-care

[10] *Why 32-bit mcus are not always the superior choice? comparing 8-bit, 16-bit, and 32-bit microcontrollers*. Components101. (n.d.). https://components101.com/article/difference-between-8-bit-16-bit-and-32-bit-microcontrollers

[11] *Advantage of java over other programming languages*. Invensis. (n.d.). https://www.invensis.net/blog/benefits-of-java-over-other-programming-languages

[12] Comparing common MCU interface protocols. (n.d.).
https://www.avnet.com/wps/portal/us/resources/article/comparing-common-mcu-interface-protocols

[13] *Embedded Software Programming languages: Pros, cons, and comparisons of popular languages*. Qt. (n.d.-a).
https://www.qt.io/embedded-development-talk/embedded-software-programming-languages-pros-cons-and-comparisons-of-popular-languages

[14] *Embedded Software Programming languages: Pros, cons, and comparisons of popular languages*. Qt. (n.d.-b).
https://www.qt.io/embedded-development-talk/embedded-software-programming-languages-pros-cons-and-comparisons-of-popular-languages

[15] Media, O. (n.d.). *JavaScript for embedded devices*. Embedded Computing Design.
https://embeddedcomputing.com/technology/software-and-os/javascript-for-embedded-devices

[16] Diachenko, B. R. (2024, March 6). Top 7 smart home protocols compared. Lemberg Solutions.https://lembergsolutions.com/blog/smart-home-protocols-explained

[17] Singh, G. (2020, February 12). Energy efficiency in Android applications. Medium. Analytics Vidhya.
https://medium.com/analytics-vidhya/energy-efficiency-in-android-application-9d1e37c55d2d

[18] Xamarin official support policy: .NET. (n.d.). Microsoft.
https://dotnet.microsoft.com/en-us/platform/support/policy/xamarin#:~:text=Xamarin%20support%20will%20end%20on,aboutCoopossum. How open-source is Android really? Medium 2022.%20upgrading%20Xamarin%20projects%20to%20

[19] Soegaard, M. (2024, April 23). Native, web or hybrid app: Which One is better? The Interaction Design Foundation. Interaction Design Foundation.
https://www.interaction-design.org/literature/article/native-vs-hybrid-vs-responsive-what-app-flavour-is-best-for-you#the_take_away-25

[20] Technostacks. (2024, January 9). Top 10 mobile app development frameworks in 2024: Technostacks. Technostacks Infotech.
https://technostacks.com/blog/mobile-app-development-frameworks/

[21] Iqbal, F. (2023, December 6). Flutter vs native: How is flutter better than native development? Medium. Zee Palm,
LLC.https://medium.com/zee-palm/how-flutter-outshines-native-application-development-c82c62fce1fc#:~:text=Native%20technologies%20allow%20developers%20to,resources%20like%20time%20and%20money.

[22] Coopossum. How open-source is Android really? Medium 2022.

Appendix B – Copyright

Prompt: Can you compare the different programming languages that are available and are most commonly used for embedded software?

Sure, here's a comparison of some of the most commonly used programming languages for embedded software:

1. C
   Advantages: Efficient and widely used. Industry estimates say 80% of embedded systems use the C programming language.
   Disadvantages: Requires developers to understand and use technical coding techniques that can be complicated.
2. C++
   Advantages: Can be as efficient as using C, but it has a standards library that can save programmers time in writing code.
   Disadvantages: A complex language that can be difficult to learn.
3. Python
   Advantages: Python is a popular programming language that's easy to learn and use.
   Disadvantages: Python may not be as efficient as C or C++ for certain tasks.
4. Java
   Advantages: Java is platform-independent, which means you can run the same code on different types of devices.
   Disadvantages: Java might not be as efficient as C or C++ for certain tasks.
5. Rust
   Advantages: Rust provides memory safety without garbage collection and has a rich type system which makes it a good choice for system programming.
   Disadvantages: Rust has a steep learning curve and is not as widely used as C or C++.
6. Ada
   Advantages: Ada is designed for large, long-lived applications where safety and security are critical.
   Disadvantages: Ada is not as widely used as other languages and may have fewer resources available.
7. JavaScript
   Advantages: JavaScript is easy to learn and use, and it's supported by a large community.
   Disadvantages: JavaScript is not typically used for low-level programming.
8. Go
   Advantages: Go is simple and efficient, making it a good choice for system programming.
   Disadvantages: Go is not as widely used as other languages and may have fewer resources available.

5.3

Prompt: I'm an electrical engineer and I have to compare all of the different technologies for temperature sensors. My main parameters are size, cost, power draw, response time, and accuracy. Please compare all of the technology for those parameters only.

ChatGPT

Sure thing! Let's break down the temperature sensor technologies based on your parameters:

Size:

Thermocouples: Small.

RTDs (Resistance Temperature Detectors): Moderate.

Thermistors: Small to moderate.

Infrared (IR) Sensors: Small.

Semiconductor-based sensors: Very small.

Cost:

Thermocouples: Generally low.

RTDs: Moderate to high.

Thermistors: Low to moderate.

Infrared (IR) Sensors: Moderate to high.

Semiconductor-based sensors: Moderate.

Power Draw:

Thermocouples: Low.

RTDs: Moderate.

Thermistors: Low to moderate.

Infrared (IR) Sensors: Moderate to high.

Semiconductor-based sensors: Low to moderate.

Response Time:

Thermocouples: Fast.

RTDs: Moderate.

Thermistors: Moderate to fast.

Infrared (IR) Sensors: Fast.

Semiconductor-based sensors: Fast.

Accuracy:

Thermocouples: Good.

RTDs: High.

Thermistors: Moderate to high.

Infrared (IR) Sensors: Moderate to high.

Semiconductor-based sensors: Moderate to high.

Consider the specific requirements of your application and weigh these factors accordingly. For example, if size and low power draw are critical, semiconductor-based sensors might be a good fit. If high accuracy is a priority, RTDs could be preferred. And if cost is a significant factor, thermocouples might be a more economical choice. Let me know if you need more details on any specific technology!

LLAMA 2

Here is a comparison of different temperature sensor technologies based on the parameters you've specified:

1. Thermocouples:

* Size: Varies (but generally small)

* Cost: Moderate to high
* Power draw: Low
* Response time: Fast (less than 1 second)
* Accuracy: High (±0.5-1.0°C)
2. Resistance Temperature Detectors (RTDs):
* Size: Small to medium
* Cost: Moderate to high
* Power draw: Low
* Response time: Moderate (1-10 seconds)
* Accuracy: High (±0.5-1.0°C)
3. Thermistors:
* Size: Small to medium
* Cost: Low to moderate
* Power draw: Low
* Response time: Fast (less than 1 second)
* Accuracy: Moderate (±1-3°C)
4. Infrared (IR) sensors:
* Size: Small to medium
* Cost: Low to moderate
* Power draw: Low
* Response time: Fast (less than 1 second)
* Accuracy: Moderate (±1-3°C)
5. Semiconductor temperature sensors (such as diodes or transistors):
* Size: Small
* Cost: Low
* Power draw: Low
* Response time: Fast (less than 1 second)
* Accuracy: High (±0.5-1.0°C)
6. Capacitive temperature sensors:
* Size: Small
* Cost: Low to moderate
* Power draw: Low
* Response time: Moderate (1-10 seconds)
* Accuracy: Moderate (±1-3°C)
7. Pyroelectric temperature sensors:
* Size: Small to medium
* Cost: Moderate to high
* Power draw: Low
* Response time: Fast (less than 1 second)
* Accuracy: High (±0.5-1.0°C)
It's worth noting that the cost and accuracy of a temperature sensor can vary depending on the specific implementation and the requirements of the application. Additionally, some temperature sensors may have additional features or benefits that are not captured in this comparison. Ultimately, the choice of temperature sensor will depend on the specific requirements of your application and the trade-offs you are willing to make between factors such as cost, accuracy, and response time.

5.2
Prompt

I would like to create a mobile app for a smart tinting window project. Please compare and select a list of software technologies to use. Provide at least 3 options.

ChatGPT

Certainly! Creating a mobile app for a smart tinting window project would require technologies that support real-time communication, user interface design, and potential integration with hardware components. Here are three options you might consider:

1. **React Native:**
   - **Description:** React Native is a popular framework for building cross-platform mobile apps using JavaScript and React. It allows for the development of native-like apps for both iOS and Android platforms.
   - **Advantages:**
     - Single codebase for both iOS and Android platforms.
     - Good performance with native-like UI components.
     - Large community and ecosystem of third-party libraries.
     - Easy integration with hardware components via native modules.
   - **Considerations:**
     - May require additional setup for real-time communication with hardware components.
     - Learning curve if you're not familiar with React.
2. **Flutter:**
   - **Description:** Flutter is Google's UI toolkit for building natively compiled applications for mobile, web, and desktop from a single codebase. It uses the Dart programming language.
   - **Advantages:**
     - Hot reload feature for quick iteration and development.
     - Single codebase for multiple platforms.
     - Rich set of customizable widgets for building UI.
     - Good performance.
   - **Considerations:**
     - Might require additional effort to integrate with hardware components compared to native development.
     - Dart might have a learning curve if you're not familiar with it.
3. **NativeScript:**
   - **Description:** NativeScript is an open-source framework for building truly native mobile applications using JavaScript or TypeScript. It provides direct access to native APIs and UI controls, allowing for high-performance and native-like user experiences.
   - **Advantages:**
     - Direct access to native APIs and UI controls.
     - Supports Angular, Vue.js, or plain JavaScript/TypeScript for development.
     - Easy integration with hardware components.
     - Good performance.

- **Considerations:**
  - Smaller community compared to React Native or Flutter.
  - The learning curve for integrating with native APIs if you're not familiar with them.
  - May require platform-specific code for certain features.

These options provide a range of choices depending on your preferences, existing skill set, and project requirements. Evaluate them based on factors such as development speed, performance, platform support, and integration capabilities with hardware components.

Copilot

Certainly! Creating a mobile app for a smart tinting window project involves choosing the right software technologies. Here are three options to consider:

1. **Electrically Controlled Window Tint with Stick-On Electrochromic Film:**
   - This approach involves using thin, self-adhesive, stick-on electrochromic film that can be applied to existing standard windows. When activated, the film becomes mirror-like and opaque, blocking sunlight and maintaining privacy. You can control it using a simple smartphone app or even automate it based on weather conditions or time of day[1].
   - **Advantages:**
     - Cost-effective (no need to replace entire windows), easy installation, and customizable control.
   - **Considerations:**
     - Ensure compatibility with your specific smart tinting window project.
2. **Sure Cuts A Lot (SCAL):**
   - Sure Cuts A Lot is a user-friendly and affordable cutting software that's commonly used by window tinters. While it's primarily designed for cutting vinyl and other materials, it can be adapted for creating designs and patterns related to smart tinting windows. It's budget-friendly and suitable for small-scale projects[2].
   - **Advantages:**
     - Affordable, user-friendly, and suitable for window tinting designs.
   - **Considerations:**
     - May require customization for smart tinting features.
3. **Tint Wiz:**
   - Tint Wiz is an all-in-one window tint app that works across various platforms (browser, mobile, tablet, and computer). It's designed specifically for window tinters and can help you manage projects, calculate measurements, and create accurate patterns. It offers a 30-day free trial so you can explore its features before committing[3].
   - **Advantages:**
     - Tailored for window tinting professionals, cross-platform compatibility, and project management features.
   - **Considerations:**
     - Explore its capabilities during the trial period to ensure it meets your smart tinting window app requirements.

Remember to evaluate each option based on your project's specific needs, budget, and desired functionality. Good luck with your smart tinting window app development!