In-Wall Smart Outlet
SENIOR DESIGN 1

University of Central Florida
Department of Electrical Engineering and Computer Science

Dr. Lei Wei

Initial Project and Group Identification Document
- Divide and Conquer -

Group 19
Miguelangel Aponte, Electrical Engineering
Eric Baez, Electrical Engineering
Felipe Caseiro, Computer Engineering
Lucas Saber, Computer Engineering

Sponsors:
None

Table of Contents

1.0 Executive Summary **1**

2.0 Project Description **3**

2.1 Motivation 3

2.2 Goals and Objectives 3

2.3 Function 4

2.4 Similar Products 6

2.4.1 Ghome Smart Mini Plug 6

2.4.2 Kasa Smart Plug 7

2.4.3 TopGreener Smart Wifi Outlet 8

2.4.4 Honeywell Home UltraPro Z-Wave Smart Outlet 8

2.4.5 Sense Energy Monitor 9

Figure 1. Resistive Load Comparison 10

2.5 Project Specifications 11

Table 1. Project Specifications 11

2.6 House of Quality 11

Figure 2. House of Quality 13

2.7 Project Operation Manual 14

2.7.1 Steps to Operate The System 14

Figure 4. Workload Divider 15

2.8 Application Description 16

Figure 5. Application Representation 16

2.9 Summary after project completion 20

3.0 Research **21**

3.1 Microcontroller 21

3.1.1 Microcontroller Options 22

3.1.1.1 Arduino Atmel Microcontrollers 22

3.1.1.1.1 ATMega328P 23

3.1.1.1.2 ATMega2560 23

3.1.1.2 TI MSP430 Microcontrollers 23

3.1.1.2.1 TI MSP430G2553 24

3.1.1.2.2 TI MSP430FR6989 24

3.1.1.3 STM Microcontrollers 24

3.1.1.3.1 STM32F103C8T6 24

3.1.1.3.2 STM32WB55 25

3.1.1.4 Expressif Microcontrollers 25

3.1.1.4.1 ESP8266 25

3.1.1.4.2 ESP32 25

3.1.2 Microcontroller Comparison 26
Table 2. Microcontroller Comparison 26
3.1.2.1 Cost 27
3.1.2.2 Memory 27
3.1.2.3 General Purpose Input/Output 28
3.1.2.4 Clock Frequency 28
3.1.2.5 Wireless Communication 29
3.1.3 Microcontroller Decision: ESP32 29
Figure 6. ESP32 29
3.2 Interface Selection 30
3.2.1 OLED Screen 30
3.2.2 LCD Screen 31
3.2.3 E-Ink 32
3.2.4 Phone Application 32
3.2.5 Screen Selection 33
Table 3. Panel Type Comparison 33
3.2.6 Smart Outlet vs Smart Plug with LED/LCD 34
Figure 7. Smart Outlet vs Smart Plug with LED/LCD 34
3.3 Buttons and Switches 35
3.3.1 Momentary Buttons or Push Buttons 35
3.3.2 Maintained Switches or Latching Buttons 36
3.3.3 Button Selection 36
Figure 8. Push Button vs Switch Button 36
Table 4. Button Comparison 37
3.4 Current Sensor 37
3.4.1 AC Current Transformer Sensor 37
Figure 9. A/C Current Transformer 38
3.4.2 Current Sensor 39
3.5 Relay Module 39
Figure 10. Relay Module 40
3.5.1 Solid State Relay 40
3.5.2 Reed Relay 40
3.5.3 Polarized Relay 41
3.5.4 Buchholz Relay 41
3.5.5 Overload Protection Relay 42
3.5.6 Inverse Definite Minimum Time Relays 42
3.5.7 Relay Decision 42
Table 5. Relay Comparison 42
3.6 Analog to Digital Converter 43
Figure 11. ESP32 Behavior: ADC Result vs Voltage Input 44
3.6.1 Analog to Digital Converter Options 44

3.6.1.1 ADS1110 44
3.6.1.2 ADS1115 44
3.6.1.3 AD7705 45
3.6.2 Analog to Digital Converter Comparison 45
Table 6. Analog to Digital Converter Comparison 45
3.6.2.1 Sampling rate 45
3.6.2.2 Resolution 45
3.6.2.3 Differential Input Channels 46
3.6.2.4 Interface Options 46
3.6.3 Analog to Digital Converter Decision 46
3.7 120V AC to 5V DC Converter 47
3.7.1 Building Half and Full Wave Rectifier 47
Figure 12. AC/DC Converter Full Bridge 48
3.7.2 Building Center Tapped Rectifier 48
Figure 12. AC/DC Center Tapped 48
3.7.3 Three Phase rectifier 49
Table 7. AC to DC Comparison 49
3.7.4 Buying the AC/DC Converter 49
Table 8. AC to DC Converter Specifications 50
3.8 Wireless Communication Methods 50
3.8.1 Wifi 50
3.8.2 Bluetooth 51
3.8.3 Zigbee 52
3.8.4 Wireless Communication Selection 52
Table 9. Wireless Communication Methods Comparison 52
3.9 App Database 53
3.9.1 Database Options 54
3.9.1.1 Firebase 54
3.9.1.2 Realm 54
3.9.1.3 AWS 54
3.9.2 Database Comparison 55
Table 10. Database Comparison 55
3.9.2.1 Cost 55
3.9.2.2 Storage At Given Price 55
3.9.2.3 Consistency 56
3.9.2.4 Reliability 56
3.9.3 Database Decision 56
3.10 PCB Comparison 57
Figure 14. JLPCB Stack 58
Figure 15. DE0701 PCB 59
3.11 Summary after project completion 59

4.0 Design Constraints and Standards **60**
4.1 Constraints 60
4.1.1 Size 60
4.1.2 Sturdiness 60
4.1.3 Cost 61
4.1.4 Satisfaction of the User 61
4.1.5 Time 61
4.2 Standards 62
4.2.1 Room Standards 62
4.2.1.1 Bathroom Standards 62
4.2.1.2 Kitchen Standards 62
4.2.1.3 Living Room, Dining Room, and Bedroom Standards 63
4.2.1.4 Stairway Standards 63
4.2.1.5 Hallway Standards 64
4.2.1.6 Closet Standards 64
4.2.1.7 Laundry Room Standards 64
4.2.1.8 Garage Standards 65
4.2.1.9 General Room Standards 65
4.2.2 GFCI Protection 66
4.2.2.1 GFCI Outlet vs Normal Outlet 66
4.2.2.2 Further Clarification GFCI Outlet 67
Figure 16. GFCI Outlet 67
4.2.3 AFCI Protection 67
4.2.3.1 Different Types of Faults 68
4.2.3.1.1 Short Circuit 68
4.2.3.1.2 Ground Fault 68
4.2.3.1.3 Arc Fault 68
4.2.3.1.4 AFCI Protection Explained 69
4.2.3.1.5 Different types of AFCI Protection 69
4.2.3.1.6 Branch/Feeder Breaker AFCI 69
4.2.3.1.7 Combination Breaker AFCI 70
4.2.3.1.8 Breaker AFCI and GFCI Protection 70
4.2.4 Cable and Wiring 71
4.2.4.1 Wires 71
4.2.4.2 Wires - Stranded vs solid 72
4.2.4.3 Cables 72
Figure 19. Wire Color Code 73
4.2.5 The Application 73
4.2.5.1 Javascript 73
4.2.5.1.1 HTML 74

4.2.5.1.2 CSS 74
Figure 20. Code Structure Illustration 74
4.2.5.1.3 JSX 74
4.2.5.1.4 XML 75
4.2.5.1.5 Linter 75
4.2.5.1.6 Github 76
4.3 Summary after project completion 76

5.0 Project Design **77**
5.1 Hardware Design 77
5.1.1 Hardware Design Methodology 79
5.1.2 Housing Design 79
Figure 21. Housing Design Reference 80
Table 11. Common Sizes for Electrical Boxes 81
Figure 22. Housing Design Example 82
5.1.3 Part Schematics 82
5.1.3.1 Power Components 82
Figure 23. AC to DC Converter Schematic 82
Figure 24. Voltage Regulator 83
5.1.3.2 Button 83
Figure 25. Button Schematic 83
5.1.3.3 Relay 83
Figure 26. Relay Schematic 84
5.1.3.4 Current Transformer 84
Figure 27. Current Transformer Schematic 85
5.1.3.5 Voltage Transformer 85
Figure 29. Voltage Transformer ZMPT101b and LM358 86
5.1.3.6 Analog to Digital Converter 86
5.1.3.7 Overall PCB Design Schematic 86
Figure 31. Overall PCB Design Schematic 87
5.1.4 Block Diagrams 88
Figure 32. Hardware Design Block Diagram 88
5.2 Software Design 88
5.2.1 Design Methodology 89
5.2.1.1 Application Flowchart 89
Figure 33. Software Application Flowchart 90
5.2.2 Development Tools 90
5.2.2.1 Version Control System 91
Figure 34. Version Control System Diagram 92
5.2.2.2 Visual Studio Code Text Editor 92
5.2.2.3 Arduino Integrated Development Environment (IDE) 93

5.2.2.4 Android Studio Emulator 95
5.2.3 Sensor Data Processing 96
5.2.4 Wireless Data Scheme 97
5.2.4.1 Shared Message Information 97
5.2.4.2 App to Outlet Messages 98
Message Type 1, Pairing Mode: 98
Message Type 2, Requesting State Change: 98
Message Type 3, Sensor Data Request: 98
5.2.4.3 Outlet to App Messages 98
Message Type 1, Pairing Mode: 99
Message Type 2, Current State: 99
Message Type 3, Sensor Data: 99
5.2.4.4 Messages Working Together 99
5.2.4.5 Final Notes 99
5.2.5 App Design 99
5.2.5.1 App architecture 100
5.2.5.2 Application Flowchart 101
5.2.5.3 Testing and Simulation 102
5.2.5.3.1 Firmware testing 102
5.2.5.3.2 Software testing 103
5.3 Summary after project completion 104

6.0 Project Prototype Testing Plan 105
6.1 Hardware Testing Environment 105
6.1.1 Current Sensor Testing 105
Figure 40.Current Sensor Testing Tight 106
Figure 41.Current Sensor Testing Loose 106
6.1.2 ADC Testing 107
Figure 42.ADC Testing 107
Table 13. ADC Readings 107
Figure 43.Analog Signal 108
6.1.3 Relay Circuit Testing 109
Figure 45.Relay Testing Circuit Example 110
6.1.4 ESP32 Wireless Communication Testing 110
6.1.5 AC/DC converter testing 111
6.1.6 Testboard Example 112
Figure 48. Breadboard Prototype Test 112
6.2 Summary after project completion 113

7.0 Administrative 114
7.1 Estimated Budget and Financing 114
Table 14. Budgeting and Financing Estimations 114

7.2 Estimated Timeline 115
Table 15. Estimated Timelines 115
7.3 Stretch Goals 117
7.3.1 Mesh Network 117
7.3.1.1 Graph Designs 117
7.3.1.1.1 Binary Tree 117
7.3.1.1.2 Non-Binary Tree with Height 1 118
7.3.1.2 Locality of The System 119
7.3.1.2.1 Local Peer-To-Peer 119
7.3.1.2.2 Internet Peer-To-Peer 119
7.3.1.2.3 Hybrid Design 120
7.3.1.3 Deciding on an Architecture 120
7.3.2 Increased Calculation Accuracy 120
7.3.3 Multi-platform compatibility 121
7.3.4 Voltage Transformer Sensor 121
7.3.5 A more secure application 122
7.3.6 A bug-free application 122
7.3.7 Testing the applications easiness 122
7.4 Summary after project completion 123

8.0 Conclusion 123
8.1 Project Alternatives 124

9.0 Appendices 124
9.1 References 125
9.2 Copyright Permission 126

Figure Index

Figure 1. Resistive Load Comparison	10
Figure 2. House of Quality	14
Figure 3. Application Representation	14
Figure 4. Workload Divider	16
Figure 5. Application Representation	17
Figure 6. ESP32 Microcontroller	29
Figure 7. Smart Outlet vs. Smart Plug with LED/LCD	34
Figure 8. Push Button vs Switch Button	36
Figure 9. A/C Current Transformer Sensor	37
Figure 10. Relay Module	39
Figure 11. ESP 32 Behavior: ADC Result vs Voltage Input	43
Figure 12. AC/DC Converter Full Bridge	47
Figure 13. AC/DC Converter Center Tapped	47
Figure 14. JLPCB Stack	58
Figure 15. DE0701 PCB	59
Figure 16. GFCI Outlet	67
Figure 17. Breaker AFCI	70
Figure 18. Wire Gauge Comparison	71
Figure 19. Wire Color Code	73
Figure 20. Code Structure Illustration	74
Figure 21. Housing Design Reference	80
Figure 22. Housing Design Example	81
Figure 23. AC to DC converter Schematic	82
Figure 24. Voltage Regulator Schematic	82
Figure 25. Button Schematic	83
Figure 26. Relay Schematic	84
Figure 27. Current Transformer Schematic	84
Figure 28. Voltage Transformer ZMPT101B	85
Figure 29. Voltage Transformer ZMPT101B and LM358	85
Figure 30. Analog to Digital Schematic	86
Figure 31. Overall PCB Design Schematic	86
Figure 32. Hardware Design Block Diagram	87

Figure 33. Software Application Flowchart 90
Figure 34.Version Control System Diagram 92
Figure 35. ESP 32 Setup Menu 94
Figure 36. Serial Monitor Output 95
Figure 37. Serial Plotter Output 95
Figure 38. Android Studio Emulator Environment 96
Figure 39. Application Flowchart 101
Figure 40. Current Sensor Testing Tight 106
Figure 41. Current Sensor Testing Loose 106
Figure 42. ADC Testing 107
Figure 43. Analog Signal 108
Figure 44. Analog Signal with Analog to Digital 109
Figure 45. Relay Testing Circuit Example 110
Figure 46. ESP32 Connection Testing 111
Figure 47. AC to DC Reference Waveform 112
Figure 48. Breadboard Prototype Test 113
Figure 49. Basic Binary Tree 118
Figure 50. Binary Tree Height One 119

Table List

Table 1. Project Specifications 11
Table 2 Microcontroller Comparison 26
Table 3. Panel Type Comparison 33
Table 4. Button Comparison 36
Table 5. Relay Comparison 41
Table 6. Analog to Digital Converter Comparison 44
Table 7. AC to DC Comparison 48
Table 8. AC to DC Buying Specifications 49
Table 9. Wireless Communication Methods Comparison 51
Table 10. Database Comparison 54
Table 11. Common Sizes for Electrical Boxes 79
Table 12. Common Sizes for Electrical Boxes 79
Table 13. ADC Readings 106
Table 15.Budgeting and Financing Estimations 113
Table 16. Estimated Timeless 114

1.0 Executive Summary

The past two years of covid have shown us the importance of being connected in this increasingly digital world. The ability to understand and communicate with each other effectively is increasing with each passing day. One area in which the disconnect is blatantly obvious is between a home owner and his power company. Every month you receive a bill for the overall usage of electricity within your house. This bill gives you no breakdown such as what has been consuming the most electricity or what rooms are consuming the most electricity. Of course each device has estimates of its yearly power consumption and how much it would cost, but estimates are not perfect, and in places with wholesale electricity like Texas, these averages end up being incorrect. Furthermore, with inflation happening in the U.S. at unprecedented levels, these averages are only as good as the day they were bought. We should not have to rely on estimates and averages in regards to our electricity consumption. What if this information was available to us at our fingertips in an easy to use system that you would be able to control. Our goal is to do this and more.

With our smart outlet, you would be able to measure the power draw of individual devices as well as set power limits for these devices within a given time frame. For example, let's say we had a gaming P.C. Due to the varied nature of gaming, the PC may require anywhere from 100 watts per hour to 330 watts per hour. You would be able to measure the exact energy usage from the P.C as well as get a warning when you were nearing a large number for the month. This would be multiplied by your Residential Electric Rate and thus you would be able to see how much money a particular device is costing you. In an apartment with roommates, you would be able to accurately compare energy usage between habitants and there would be no discussions regarding a mini fridge causing the electricity bill to go up a huge amount since it was plugged in.

Another feature of the smart outlet would be the ability to turn it off and on from the app. This would most likely be done through a relay. The user would be able to target specific outlets within his home instead of turning off the whole room through the breaker switches. This would also work in tandem with power limits. Take for instance the gaming P.C from before, if needed you could set an overall power cap for the P.C. based on usage or cost, and turn off the outlet. In locations with wholesale electricity, this would allow you to put a large cap on the outlets that if reached would save you a large amount of money. During the winter storms in Texas, electricity rates jumped up to 3.00 dollars per kWh. This led to customers paying excessive amounts of money without even knowing and without having any breakdown information on the energy usage. With our device, a user would be able to set a money limit such as $500 in which non essential devices would be powered off immediately to save cost. Now $500 is already a lot of money to pay for a bill but it is significantly better than the thousands some customers had to pay when rates spiked. Overall, a device like ours gives consumers knowledge and knowledge is power.

Electricity usage and monthly cost would be available through a companion application. Logs of usage throughout the month would help bill payers understand where their money is going. Furthermore, the user would be able to individually turn off each smart outlet from the app in desired situations, such as going on vacation.The goal is to allow the consumer access to specific information regarding their home instead of just having a single amount that they have to pay every month.

Currently there are a few other products on the market but there are only a few in wall smart outlets and those that do exist do not have all of the features we are trying to implement. The best in wall smart outlet according to the New York Times, is the Top Greener In-Wall Smart Wi-Fi USB Charging outlet but that one only has one controllable receptacle where-as we would like to be able to control both of the outlets. Our generation has shown a desire for everything to be minimal while having a visually pleasing aesthetic. Smart outlet boxes that go over the outlet create visual clutter and take up space in an economy where space is a premium. By having the outlet flush with the wall, you create a more cohesive environment as well as maintaining the usable space in your home or dwelling.

The market for this device would be everyone in a typical household. This includes anyone from homeowners to college students in a dorm. Measuring energy consumption is a vital tool that can be used by anyone. I know from personal experience how frustrating it is living in student housing, being told you went over the monthly electricity cap and being forced to pay extra money without having any information on what device used that extra power.

The Smart Outlet is meant to add power and knowledge to the consumer side instead of just having another bill to pay each month. Why should we not be able to break-down our monthly energy expenses into the devices or location we desire?

2.0 Project Description

Below are descriptions for our entire project. This will include a description of the project itself, why we wanted to do it, and why we think the project is meaningful from an engineering standpoint. It also outlines other key descriptions, such as similar products, project specifications, and an operation manual for easy access.

2.1 Motivation

Our biggest motivation for this project is cost awareness. All over the world, people have to pay for electricity for all of their household electronics. However, in this digital age, people are starting to use more and more electronics at once in their homes. As a result, many people find that their electric bill becomes increasingly difficult to keep track of. With so many devices, it becomes harder to track where extra power use is actually coming from. As college students, all of the members of our team have experienced going over an apartment’s monthly cap for electricity usage, which requires you to pay an additional fee. In this case, it is very difficult to reduce your usage since you do not know which roommate is using which device too much. From college students to anyone else in the world, nobody wants to pay more than they need to for electricity. If you knew how much power each device in your home was using, you could easily figure out a plan to stop using that device as much. The goal of our Senior Design project is to develop a smart outlet with a companion app that can help someone be aware of the power they are using per device. With this system, we want to make both managing and keeping track of power used by an outlet easier, which can help people to understand their electricity bill better and even save money.

2.2 Goals and Objectives

When our group first started this project, we had plenty of ideas going around of what type of product we were trying to create. After plenty of good pitches, we finally settled on one that was of interest to all members of the group: an outlet capable of measuring how much power a user is utilizing every month. In addition to this, we wanted to create an application that would be capable of analyzing these numbers and give a more detailed insight into how the users are using their electricity.

We knew this project would be challenging, since none of us had any experience working with high voltage devices like this, but we took on the challenge nonetheless. Also, some of us had experience with app design in previous classes, but the application we plan on developing is on a much larger scale, that will require a bluetooth connection to the electrical device and many cybersecurity protocols.

Speaking of cybersecurity, that is one of our major goals for this project. One of our group members has taken some classes on the subject, and can definitely implement the knowledge developed from those courses into our project. For the application, we plan to introduce a two-step verification system, which will require the user to insert their phone number into the application and confirm the veracity of their identity via a code sent through text message. Besides that, we also plan to implement captcha verification in our application, trying to prevent hackers from brute forcing their way through figuring out a user's password. These implementations will not be easy to insert into our project, but since it’s one of our main goals, it is important for us to find as many articles as we can on the subject, so that we can achieve this goal.

Another one of our main goals is to make progress in this project as smoothly as we can, so that none of our group members have to stress over meeting deadlines or being stuck in a part of the project. To achieve that, we have been meeting on a weekly basis, keeping each other updated on what we have achieved during the week. In Senior Design I, this means talking to each other about the project we made on the document, and what parts of the project we should focus on next. We also plan to tackle everything in groups of 2, so that none of us has to tackle a part of our project alone and be stuck because of it.

The main goal we have in this project, however, is make people more conscious about their electricity usage throughout the day. The average American,according to a study done in 2020, uses about 893 kWh of electricity per month, which averages out to about 37.2 kWh of electricity every day. By giving the user a way to visualize how much energy they are using in a single outlet, we hope that they will be more conscious about their energy usage, and will hopefully reduce their daily usage.

2.3 Function

For the proper functionality of our project, first you’ll need to install our Smart Outlet in our home’s outlet. In order to do that, a simple installation process needs to be done by the customer, by simply plugging in the Smart Outlet on the outlet you desire to measure the voltage. After that, the customer will check if the product is emitting a light, indicating that it is on and that it was installed properly.

Before the customer starts connecting to the device, they must be registered and signed into the device’s application. The application is essential for our project, since it’s where most of the information is going to be displayed. This creates a constraint in some cases, since not all users have access to a newer phone (this will be elaborated on later). After doing so, the user will then be able to fully enjoy all the functionalities of the Smart Outlet without restriction.

To give the user an easier time, we will implement a function for them to change the name of the device in their application. This is gonna serve the purpose of allowing the user to set their device name to whatever they want, making it easier to identify which device they wish to control. For example, a device located in the Living Room could be renamed to “Living_Room” or, if there are multiple devices within the same room, they could be named “TV_Device” and “Computer_Device”. The user will have complete control over the name, as long as it doesn’t include spaces and has too many characters. In case the user tries to name a device with spaces or more than the allowed amount of characters, the application will emit a warning.

If the Smart Outlet is functioning properly, it should already start measuring out the voltage and current. Next up, the customer will have to link their device to their mobile phone number by enabling the bluetooth in both of these devices. First, the user will have to enable the bluetooth on the Smart Outlet, simply by pressing a button in their device. If the light tagged as “Bluetooth” is blinking at a constant speed, it means that device is ready to be paired up. The customer will then have to connect to the device in their mobile phone, via the application. The application will then emit a warning, indicating that a new device has been registered to your account.

After following all these steps, the Smart Outlet should be ready to be used at its full potential. With the application properly set up and the Smart Outlet in place, the device will start calculating the voltage and current and sending the information to the customer’s mobile phone device. With the application, the customer will have access to all of the devices functionalities unrestricted, such as turning the device on and off, setting a timer for the device, and checking out the monthly usage for a specific device.

When the customer wants to simply turn on the device, all they’ll have to do is click a button on the application, and the device will instantly start supplying power through the outlet. In order to do the inverse and shut off the device, the user will have to click the very same button (the interface will show if the device is on or off when the user chooses that option). The device will then instantly stop supplying power through it.The user can also choose which of the connected devices they want to turn on oroff, and the application will display a list of all the devices, so that the user can choose which one they prefer. For this option to work, however, the user will have to be in range for the smartphone to detect the device via Bluetooth. In case the user is too far away from the device, this option will not work, and the application will emit a warning that there is no Smart Outlet in range.

When the customer wants to set a timer on the device, he will have to choose a valid timeframe within the application, and then press the button to start the timer. As soon as the timer hits 0, the device will instantly shut off power through it, and will have to be manually turned on before the power supply starts again. The user can also choose which of the connected devices they will set a timer to, and the application will display a list of all the devices, so that the user can choose which one they prefer. For this option to work, however, the user will have to be in range for the smartphone to detect the device via Bluetooth. In case the user is too far away from the device, this option will not work, and the application will emit a warning that there is no Smart Outlet in range.

The last option is when the user simply wants to check their current power utilization. This process will involve only the application, and the user won't have to be in range of the device for it to work properly. Whenever the user clicks on this option, they will be given a detailed portfolio on the outlet's power utilization over a span of days, weeks, or even months if the user desires. The user will also be given the option to check the utilization for a single device or for a group of them, by simply selecting or deselecting each device they wish to check.

2.4 Similar Products

As stated before there are some products on the market that have similar functionality and are also very efficient. However, the most important aspect to our product is the fact that we want all of the features necessary for a smart device as well as a power measuring device rolled into one.

Another aspect we want to focus on is design aesthetic. There are quite a few smart plug devices out there on the market. The issue with the smart plug devices is that they fail in a core element- design. Clean sleek design is one of the most important aspects in the modern world. This is seen through the company Apple and their aesthetic which redefined an industry. Overall the goal is to make it simple for the consumer in regards to visuals and usage. Keeping it simple while hiding complex hardware under the hood. The smart plug fails due to protruding from the wall and causing visual clutter.

On the other hand, other smart outlets lack the power measuring that we would like to focus on for the end consumer. Some of the products that we will be comparing are: GHome Smart Mini Plug, Kasa Smart Plug, Merkury Innovations Smart Plug, TopGreener Smart Wifi Outlet, Honeywell Home UltraPro Z-Wave Plus Smart Receptacle, Sense energy monitor.

2.4.1 Ghome Smart Mini Plug

The first competitor is the Ghome Smart Mini Plug. This is a circular plug that connects on top of the outlet. It has its own app and works with the Alexa and Google Home assistant. This device works through 2.4 GHz wifi which is a simple enough method of communication between the internet and the plug.Although this could be considered a competitor the design is different than ours as we are focusing on the wall in the outlet segment. The issue with this design is that it ends up being visually cluttering especially when working with multiple of these devices. Having cylindrical bumps from the outlets to the devices can be a little awkward.

Furthermore, it has no energy monitoring aspects related to it so the most you can do is turn it on and off. This is useful for basic things such as lights, lamps, or coffee makers, but will not provide value if utilized with a computer desktop or air fryer where electricity consumption is more prioritized. Overall this generic smart plug showcases the issues with all the cheap smart plugs. The goal is to turn it on and off by connecting it to the home wifi and with the use of an app.

2.4.2 Kasa Smart Plug

Another contender is the Kasa Smart Plug. This product looks almost like a power brick that is plugged into the wall but is oriented horizontally so as not to cover the other outlet. It uses an app to be controlled easier and can also be connected to Amazon Alexa or Google Assistant to be used hands free. For example, if a lamp is connected to the smart outlet, someone could tell “Alexa” to turn off the lights and the outlet would be switched off which effectively turns the lamp off. Along with voice controls, the Kasa Smart Plug can also do what they call “smart actions.” The smart actions are interactions the smart plug can do in conjunction with other smart devices. First, the person must own the Kasa smart camera, Kasa Smart Plug, and a lamp that is plugged into the outlet. Then, they can set it up where if the camera detects movement, The smart plug can be triggered to turn on which then turns the light on. This can be especially useful if getting home late and the person’s hands are full while it is dark. That would be one less thing to worry about.

Also with that, the Kasa Smart Plug can use timers so that when the lamp triggers on, after 10 minutes, the plug would switch off and thus turn the light off. This is useful so that way the person doesn't have to remember to turn the lights back off. The timer also goes hand in hand with the schedule feature that it has. The smart plug can be scheduled to turn on at certain times of the day and turn back off via the app. Another cool feature this device has is having an “away mode.” This is for when someone goes on vacation and maybe is afraid of someone else breaking into their house. They turn this mode on and any lights or appliances that are plugged into the smart outlets will randomly power on and then power off. Finally, the Kasa Smart Plug also keeps track of run time. This is supposed to help users cut back on energy consumption if they are overusing certain devices unnecessarily.

The Kasa Smart Plug does a lot of the same functionalities that we hope to achieve with our smart outlet. It utilizes a mesh network which we are considering doing with our product depending on how seamless we can make it. The plug makes use of 2.4GHz Wifi. The device also has a physical on and off switch like we would like to have on our smart outlet. However, we do not aim to be compatible with voice controls and we aim to help users cut back on energy usage but by actually telling the user how much power they are operating with. The last key difference is that our smart outlet is an actual outlet that is in the wall whereas the Kasa Smart Plug plugs into the outlet.

2.4.3 TopGreener Smart Wifi Outlet

The TopGreener Smart Wifi Outlet is a jump in quality and price over the cheaper smart plug and outlet options available. Like other smart outlets it utilizes 2.4 GHz connections in order to communicate with the internet. These outlets utilize 2.4 GHz connection as opposed to a 5 GHz connection as a 2.4 GHz connection travels further than a 5 GHz connection but at a slower speed. When working with data like this, confirmation that data is able to be transmitted is more important than speed as opposed to other consumer devices like a phone or smart T.V. With the 2.4 GHz connection comes a mobile application to go alongside it. Furthermore, the outlet works with Alexa and Google Assistant and this makes it more convenient for consumers.

The Outlet utilizes a 15 Ampere relay in order to control the electricity connection to the connected device. A big difference from cheaper outlets is that it has a manual method of turning the outlet on and off in the form of a button on the face of the device. This is an important facet of a device like this because despite our best efforts to remain connected to the internet, outages happen or there might be connection issues that arise with an outlet you won't be able to easily remove the outlet to turn on the product if the relay is off and experiencing connection issues.

Furthermore, this is the first outlet with energy monitoring albeit only the bottom outlet of the 2 outlet device is actually able to do energy monitoring. This is actually a good method as there are some devices and appliances that must always be on. However it does limit the goal of being able to monitor every device you own. With the energy monitoring you are able monitor energy usage over time with the app and create schedules based on your needs.

2.4.4 Honeywell Home UltraPro Z-Wave Smart Outlet

Another tier up in price is the Honeywell UltraPro Z-Wave Plus Smart Outlet. The goal of Honeywell is the ecosystem of products that it owns can all be controlled from its app. Devices such as light bulbs, security cameras, and switches can all be controlled from its ecosystem so this is trying to target a consumer who wants everything connected under one umbrella. The issue with this is that in order for the smart outlet to work it requires a Z-Wave Hub in order to function in its intended way. If you already have the Hub from other devices this is fine but if not, the hub itself can range from a hundred dollars to five hundred which can add a lot of expense to a consumer not ready to take such a heavy plunge.

In other aspects, the Honeywell smart outlet behaves much the same as the TopGreener smart outlet without the energy monitoring aspect. It only allows you to control the bottom outlet from a mobile application. There appears to be a button to control the inner relay for the bottom outlet however there is no mention of this button in the specs about what it does so there is a possibility that this outlet is only able to be controlled from the application. In that case the only true selling point of this product is the Z-wave connection. The outlet has a built in Z-wave Repeater and Range extender which if you are using Z-wave Honeywell devices could be useful.

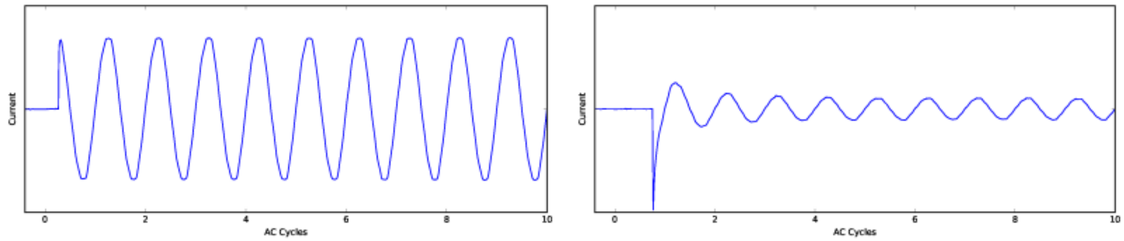
2.4.5 Sense Energy Monitor

The last of the competitors that we considered is the Sense Energy Monitor. This is a design that we considered strongly in the beginning. The difference with the designs is in how it detects the power consumption. Our smart outlet is on site where it is plugged in and detects the power drawn to the device that is plugged in. The sense energy monitor plugs into the circuit breaker with current sensors to detect the current flow into each area of the house. The power for the device is also connected into an open breaker on the circuit breaker since it will need a constant power source to function. It also has an antenna that is plugged into it so it can connect to the internet since it uses a smartphone app. Without the smartphone app, there would be no way to read the data since it does not have a display. It looks almost like a small battery pack that sits in the breaker box. The smartphone app, similar to our product, makes it much easier for the user to understand what is going on with their power consumption. The Sense also has the ability to be compatible with Amazon Alexa which is a great feature to have as homes are becoming smarter by the day.

The next question that might be asked is that if the energy sense monitor is plugged into the circuit breaker, how does it know what is using what? Well it overcomes that problem by being able to detect different electrical devices electronic signature. Basically, it detects what device is on using many different signals but the two main ones are through the breaker by how much power it uses and its resistive load. So according to sense, if a toaster in your house uses about 1200 watts, Sense will detect that the breaker is drawing 1200 watts. From there, it will tell you the toaster is on. As soon as the circuit breaker stops drawing those 1200 watts, it will know the toaster is off and Sense will tell you that the toaster turned off. However, once the toaster is already hot and ready to go, if you toast another piece of bread, it uses less power since it is already hot and does not need to heat up again. This makes measuring devices from just the power usage very tricky. Not to mention the fact that the microwave might use the same amount of power as the toaster. This is where measuring devices from its resistive load come into play.

The first graph (on the left) is the resistive load of a toaster and the second (on the right) is the resistive load of an incandescent light bulb. (Taken from https://blog.sense.com)

Figure 1. Resistive Load Comparison



As can be seen from the two graphs, the resistive load of the toaster is much different than that of the incandescent light bulb. This helps tremendously in differentiating between different types of electronic devices. Even different types of light bulbs have different resistive load graphs. To this day, the Sense team is still mapping out which data signatures belong to which types of devices. From this aspect, making this concept seemed much more complicated than initially expected which led us to settling with a smart outlet instead. Another key difference that led us to our decision is the fact that the sense does not have the ability to turn on or off devices. It will tell you if something is on but you will need to go to the outlet yourself and unplug or turn off the electrical device such as a toaster.

A few great things it does as a result of its functionality is let you know when water leaks occur. Because it can determine when devices turn off and on, once it realizes your water never turned off for a week, the user can fix that problem. If the user always leaves the A/C low when they leave for the day it can help them realize they need to turn up the temperature. Also, it can let the user know if the fridge was on for longer than it was supposed to be. Since fridges work by periodically turning on and off, if it is on longer than it should be, then the fridge needs to be fixed. All these and more have been said in many testimonials for the product on its website.

One downside that Sense does not talk about too much on their website is how it detects “12 devices in the first month and 25-30 devices after 12 months.” While 12 devices may be all you need to read, if you live in a large household with many people this number can add up quickly. Since the process it takes to figure out which device is which from its electrical signals is very complex and can take some time, this can definitely be a deal breaker for some people.

2.5 Project Specifications

Below is a table showing our project specifications. These are important because this shows what we want our project to do. Requirements 1,9, and 11 are the most important specifications.

Table 1. Project Specifications

	Requirements	Unit
1	Shall measure current accurately	±250mA
2	Shall track power usage over time	Accuracy 90%
3	Shall visualize power data over time in the app	Up to 3 Months Detailed
4	Use Minimal Power	<1200 Watts
5	Shall indicate on/off state with an LED within a certain time	4 Seconds
7	Shall fit into existing outlets without modifications needed	3.5 inches wide by 5.25 inches tall by 5 inches deep
8	Shall keep accurate measurements no matter the temperature	<120F
9	Shall turn power on/off remotely	<2 Seconds
10	Shall connect to multiple smart power outlets at the same time	Up to 5 Devices
11	Shall have a constant refresh of the power usage	30 seconds
12	Shall visualize power data over time in the app	Up to 3 Months Detailed
13	Shall set a timer to turn off outlet	Up to 24 Hours

2.6 House of Quality

Figure 2.6 shows the different correlations between each of the different technical and user components. It essentially explains that the better we want our product to be, the more money it will cost. Size just means the size of our outlet in the wall. If we want to make it as compact as possible and fit in the wall like every other outlet does, we may need to spend more money to purchase parts that are more compact. The size has nothing to do with the prongs of the outlet. The easy to use section just means how easy it is for a user to purchase our product, install, and use it with the app. The goal is for it to be intuitive so that people will use it when they buy it and not just return it. Also, since technology is very confusing nowadays, it would need to be understandable to the older market since they are the ones who pay the bills anyways and not a child who could probably understand it a lot easier.

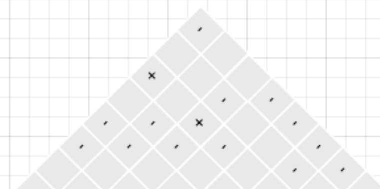
Next is the reliability, we want our product to be both durable and accurate. We want to use precise components that give accurate measurements while also being tough so that they will not break or blow up under normal circumstances. Achieving high reliability will certainly increase costs but that is one of the most important aspects of our device. We then have the smart phone app. This is closely related to ease of use in the fact that we want to make it simple and easy to use. Part of the reason we are using a smartphone app is to make our product easier to operate and understand for everyone. The power consumption will pop up on your phone app and will be straightforward to turn on and off the device. The user interface is closely related to the app as well. If we have a great user interface on the smartphone app, everyone will be happy and want to use the app. If it is a nightmare to use and look at, no one will want to use it and essentially make our product useless.

Finally for the user requirements, we have power use reduction. The whole point of our product is to limit the power consumption each person uses on a daily basis in the household. Our app in conjunction with the outlet is designed to let the user know how much power is being drawn from the outlet so that they can be more conscious of it and adjust accordingly. For example, if the user is operating one of our plugs with a mini fridge connected to it, the app will tell them how much power is being drawn. Maybe the user only has a few drinks inside the mini fridge and realizes the power required to utilize the mini fridge is not worth it. Thus, they either cut the power to the plug via the app or just simply unplug the mini fridge and move the few drinks to the kitchen fridge. Some of the engineering requirements such as LED light, toggle switch, and WIFI/bluetooth capability are to make the life of the user easier as well. The LED light would make it easier for the user from a glance to be able to tell if the power is on or off on the plug. The WIFI/bluetooth capability is a must requirement if it is to be able to connect to the phone app. That is a key feature so WIFI/bluetooth capability will be necessary no matter the cost or size. The last of those three is the toggle switch, sometimes if the smartphone is not within reach, an easier method to turn off the plug would be to simply have an off and on switch. To conserve space on the plug, the switch will most likely just be a push button since a slider button would take more real estate on the outlet.

Finally, the last two components that will be discussed are the power rate and the plastic case. It is also important that our smart outlet does not consume too much power to function because then it will be adding more power usage to the household and the whole point is to save power and money. We want something powerful enough that will accurately measure the outputs while also not being so powerful that it requires high energy to function. Lastly, the plastic case is our final engineering requirement that requires discussion. Although metal would be a more durable case, that means it would also conduct electricity which would not be great for anyone who wishes to remain functional. For that reason and for cost, a plastic casing is a no brainer. The goal of our product is to be easy and cost effective for the user and as long as we follow these requirements, that will be easy to achieve.

Figure 2. House of Quality

Key	
↑↑	Strong positive Correlation
↑	Positive Correlation
↓	Negative Correlation
↓↓	Strong negative Correlation
	No correlation
Positive Polarity	+
Negative Polarity	-



Engineering Requirements

User Requirements		Power Rate	Cost	Power Use	Sensor Accuracy	Wifi/Bluetooth Capability	LED Light	Toggle Switch	Plastic Case
		+	-	-	+	+	+	+	+
Cost	-	↓↓	↓↓	↓	↓↓	↓	↓	↓	↓
Size	-	↓	↓	↓					
Easy to use	+					↑↑	↑	↑	
Reliable	+		↓↓		↑↑				
Smart Phone App	+		↓			↑↑		↑	
User Interface	+		↓			↑		↑	
Power Use Reduction	+			↑	↑↑			↑	
		<1200Watts	<$500	<1200Watts	<20Watts	100ft Range	<1ms	<1ms	<$20

2.7 Project Operation Manual

Our system consists of an enclosed smart outlet device that can be installed onto any wall outlet in order to bring power usage awareness to its user. In order to make the information calculated meaningful to the user, we also have an app that takes sensor data from the smart outlet and converts it into a more readable form. This includes power average estimates, instantaneous power usage estimates, graphs to show usage over time, and other useful metrics. Having all of this information at the user's fingertips will allow them to easily see how much power they are using and save money in the best way possible. All of these will be available on the app's user-friendly dashboard, along with some other control features.

One of the control features will allow the user to set some preferences in order to automate the system. These preferences include an energy cap and time period for the energy cap to take place over. One of the other control features is a button that has multiple functionalities. When pressed with two short clicks, it manually toggles the smart outlet on or off. A double click is preferable here since it can help prevent accidentally bumping the button and turning the outlet on or off. The indicator light on the front face of the device will indicate the current status of the outlet. Additionally, when the button is held down for several seconds, it will initiate pairing mode, which is shown by the indicator light blinking.This is especially useful for managing power in your home since your devices won't always need control based on a pre-set limit. Sometimes, the outlet may need to be turned on or off as needed by the user. This will allow you to have control over your outlet even if the app connection cannot be established. The last primary control feature will be the ability to pair to other smart outlet devices. Pairing to other outlets will allow the user to have more of these devices in one network, enhancing their awareness of their power usage.

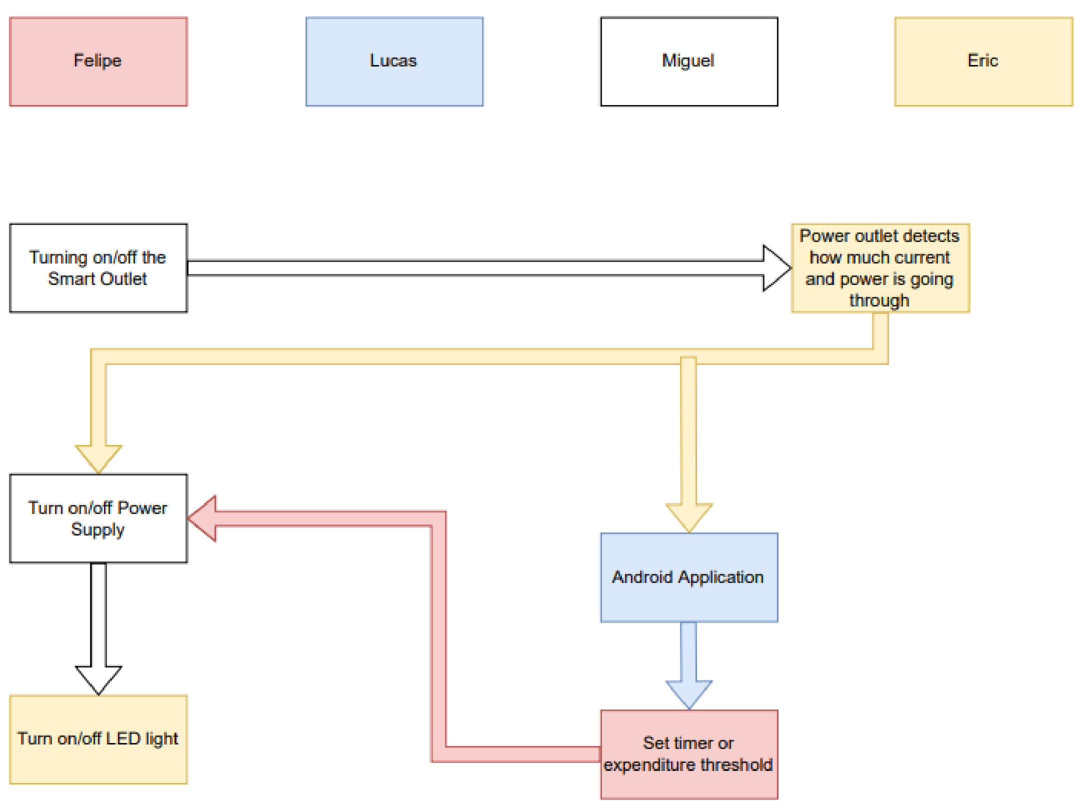
2.7.1 Steps to Operate The System

To operate this system, the first thing that needs to be done is installation. Due to size constraints, our device will be one that can be plugged into an existing outlet in your wall. To install the device, simply plug it into the outlet whose power you want to monitor. You can test the device to make sure it works by pressing and holding the button on the front to toggle the outlet on or off. You should see the indicator light on the front of the device turn on to indicate success.

The next thing to install is the app. To do this, download the app from the provided link and perform its initial setup. The most important thing to do here is to pair an outlet to it. In order to do this, you need to press the pairing button on the front of the outlet. This is easily accessible on the corner of the face by a small hole. The button is hidden in the hold in order to prevent accidental re-pairing. To press the button, insert a paperclip end into the hole until you feel a click. Hold the paperclip in place until the light on the faceplate begins to blink, indicating that it has entered pairing mode.

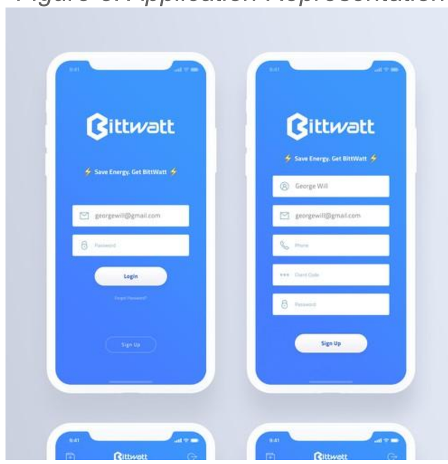
Once you put the device in pairing mode, you can then navigate to the app. When you open the app, you will first be prompted to login. If you do not have an account, there will be an option to sign up. This will require you to fill out several fields, including the email field. Once you submit all of that information, it will send an authentication link to your email. Once you click this link, you will be able to log into your account. After logging in, you will have access to the app's many features. One of these is the pairing mode feature, which will allow you to detect smart outlet devices. Since the device is in pairing mode, it should show up on the list of available devices in the app. Simply select the device and confirm that you want to pair. After that you will be redirected to the user dashboard upon pairing success.

Figure 4. Workload Divider



2.8 Application Description

Figure 5. Application Representation



(How we plan for the application to look like)

For this project, we'll be designing an application in order for the user to remotely control their smart outlet, without having to actually be in the same room as the outlet. This app will be programmed in Javascript using the react native framework and will involve all the members in the group.

Firstly, the user will be greeted by a welcome message when they first open the app, an animation with the phrase "Smart Outlet" will be shown on the center of the screen. After that, the user will be directed to a login page, where they will be given 2 options: Sign in and Create a new account.

The sign in option will open 2 user prompts: one to type in your account email and one to type in your password. The user will have to correctly type in both in order to sign in to their account. This part of the application will have clickable text to reset your password, for users who can't remember their password. In order for the users to be able to reset it, however, they will have to provide the email the account was associated with and reset their password from their email account. If the user correctly types in their account name and password, they will be directed to the main page, where they will be allowed to check all of their account information. This page will also make an option available for the user to stay signed in, so that the user won't have to type in their credentials every time they wish to check their account, giving the user an easier time and incentivizing them to check in whenever they want without a hassle. If the user is signing in from a new device, however, the user will have to type in a 6 digit code that will be sent to their email, making it harder for hackers to access someone else's account.

The create account option will open 4 user prompts: One will be for typing in the account name, where the user will be prompted to enter a unique username. In the case that the username is already taken by another user, the application will display a notification for the user to keep changing it until they can find an username that is not being used. Another one will be for typing in the email the user wants to receive all the associated emails from the app. In the case that the user types in an invalid email, the app will display a notification and it won't go away until a valid email is put in. Another one will be for the user to type in the password for the account. The password will have to have at least one capital letter, one number and one special character. If these requirements are not met, the app will display a warning until the user correctly enters a password that fulfills all of the requisites.

Another one will be for the user to re-enter their password. The password will have to be re-entered exactly as it was the first time around, and the application will display a warning until the passwords match. Finally, the last prompt will be for the user to enter their phone number. If the phone number has too many or too little digits, the application will emit a warning until the user correctly enters a valid phone number. After all of these requirements have been met, the user will be sent an email, where they will have to confirm the account creation. After the confirmation, the user will be finally able to access their account and link their designated outlets.

When the user is finally signed in, the application will present the user with a variety of options for the user, such as: Set a timer, check your monthly utilization, turn power on/off and account settings. Each of these options will make the smart outlet perform a different action, without the user having to physically interact with the outlet. The order that these options appear in the app is not yet final and can be changed as we continue to develop the application and find better solutions.

The "set a timer" option will provide the user with the interaction of being able to set a timeframe of how long they want their outlet option on for. Whenever the user clicks on that option, they will be presented with a prompt of how many minutes or hours they want their outlets on for. Whenever they select the desired timeframe, the timer will start going down, and provide the necessary power output. When the timer is done, the outlet will emit a warning to the user, indicating that their Smart Outlet is about to be turned off, and the device will instantly shut down and stop providing power. The user will be able to cancel the timer at any time, and the action will not be performed, continuing to provide power as normal. The maximum timer the user will be able to set is 3 hours and the minimum will be 1 minute. Trying to set a timer too high or too little will not be allowed, and the application will display a warning, not allowing the user to go through with the action.

The “check your utilization” option will provide the user with detail how much they have spent over the past months, since they started utilizing the Smart Outlet. For this part, we plan to implement a series of graphs, in order to give a visual representation to the user, making it more clear for the user how their use has increased or decreased over the period they’ve had the Smart Outlet. For this part, we plan to include both the cost graph and the power utilization graph, so the user can pick and choose as they wish.

The “Turn on/off” option will provide the user with the feature to either turn on or turn off the Smart Outlet, without actually having to physically interact with the outlet. This feature, although, will have a sort of “cool off” period, so that the user is not able to repeatedly turn on and off the outlet and break the device. This timer will be reset every time the button is pressed, and will emit a warning (while displaying the timer until the next possible utilization) for when the user tries to utilize this feature multiple times in a row.

The “Account Settings” option will provide the user with the capability to perform various changes to their account, if they wish. The first option will be “change your password” and will enable the user to change their current password. The second option will be to link a new Outlet, where users will be able to link a Smart Outlet they own to their account. The third option will be to unlink a Smart Outlet, where the user will be able to unlink a Smart Outlet they don’t own anymore. The last option would be Signed out, which would disconnect the account from the application. The order that these options appear in the app is not yet final and can be changed as we continue to develop the application and find better solutions.

The “change your password” option will enable the user to change their current password, and will send the user a link in their email, so that they can change their passwords, as long as it includes a capital letter, a number, and a special character. The user will also not be able to change their password to one they already used, and the application will display a warning indicating so, not going away until the password is changed into one that has not been used before. If all of these parameters are met, the password will be changed and the old password will not be able to be used to sign into the account.

The “Link your device” option will enable the user to connect an available Smart Outlet to their account so the application can start calculating the user’s power use. If the Smart Outlet is already linked to an account, the application will not go through with the action, and will display a message saying “The device is already linked to the account”. If said account unlinks the Smart Outlet, it becomes available for linking up again. It’s important to say that, to link the Smart Outlet, the device must be in the near vicinity, and will link the user’s phone via Bluetooth. If the phone is too far away from the Smart Outlet, it won’t be possible for it to link to the phone. If there are no Smart Outlets in the near vicinity, the application will display the error message “No nearby Smart Outlets, please make sure you’re in range for the pairing”

The “Unlink your device” option will enable the user to unlink a Smart Outlet from your account. Unlike the previous feature, the phone won’t have to be in range of the Smart Outlet for it to be unlinked, and the application will display a list of all the connected devices for the user to choose to disconnect. When the user unlinks a Smart Outlet from their account, the device becomes available once again for pairing. If there are no registered devices linked to the account, the application will display the following message: “No linked devices to your account, please link a device first”.

The “Sign out” option will enable the user to disconnect their account from the application, bringing them back to the front page, where they will have to effectuate the login once again or create a new account.

Another one of the main utilizations of the application will be to display the total cost of the user’s energy utilization. To do this, the user will have to enter how much their energy provider charges per kW. Using this information, the application will be able to calculate the total cost that the smart outlet is responsible for. In the application, the user will be able to check how much their cost has been every month since they signed up for Smart Outlet. With this, the user will be able to be more conscious about their energy use.

One of the main concerns our group will have with this project will be about cybersecurity. To ensure that hackers will not be able to access your information, we will implement several protocols, such as: Captcha forms, to make sure that no bruteforce programs can be used to login into a user’s account. These programs keep trying all possible combinations possible for passwords until it finds the correct one. These Captcha forms introduce a protection against such programs, by introducing a verification against bots, where a human would have to individually perform an activity every time the program got it wrong, rendering the program useless.

Another security protocol we will introduce will be a verification code through the user’s phone, where whenever they are signing in from a new location, the application will require the user to insert a code that they received through their phone. With all these protocols in place, we hope that we can create a safe application for our users, where they won’t have to worry if their financial information can fall in the wrong hands, or even if the utilization of their small outlets are being controlled by someone else.

2.9 Summary after project completion

For this part, nothing changed after the conclusion of our project. Our goals and motivations stayed the same and for our specifications, which remained about the same for our finished project. The only change comes from application description, with options for account creation slightly changed in the finished product.

3.0 Research

In order to fully understand our project as a whole, we need to first understand each of its individual parts. To accomplish this, we have several sub-sections below that analyze each of the most important parts of our project. This will allow us to come to an educated conclusion about what the best options are for our project. This will also make purchasing items much easier, since each section should come to a definitive conclusion about what technology or device we need to use for each core part of the system. The analyses below will discuss how the metrics and points of analysis for each section factor into the overall decision. Each should explain why we need that particular technology or device, as well as other relevant information about it. Additionally, each should compare different items on the market in order to narrow down options, which should yield an informed decision about the best option.

The main topics that we intend to cover in this research section include all of the different parts that we will need to purchase. The first and most important one is the microcontroller, which will serve as computing power for the outset itself. The next is the interface selection, since that determines how the user will be able to control the device. We also have the AC current transformer sensor, which is how we plan on collecting the current usage data from the outlet. We will also have the relay module, which is how we plan on having the ability to switch power to the device on and off. Next is the analog to digital converter, which is how we plan on accurately reading the data from the AC current transformer sensor. We also have the 120VAC to 5VDC converter, which is how we can adapt the house's power into a safe, usable amount for the microcontroller and other electronics. The last topic includes the different wireless communication methods, which is how we can make the device smart. This will allow us to connect to the smartphone app wirelessly, which is critical for the project to work.

For this project, we decided to leave out the standard outlet and custom PCB from the overall analysis, since they don't really have reasonably comparable alternatives. Though there are different PCB manufacturers, most of them have essentially the same results. This can also be said for the standard outlet, since different brands do not vary much. This is because the outlet we want to use is standardized, so the design is essentially exactly the same when comparing multiple different companies or vendors.

3.1 Microcontroller

The microcontroller is one of the most important parts of this entire project. It acts as the main processing power that interprets and manages data. Specifically, it reads the current transformer sensor with the ADC (Analog to Digital Converter) and runs the readings through various calculations to figure out the power being used at that instant. It also averages that information in order to account for small fluctuations and noise to get a more accurate reading of the power. It then takes that information and sends it over bluetooth to the app. Additionally, it stores a state for the outlet being on or off. It uses this state to set one of its GPIO (General Purpose Input/Output) pins to either high or low. This is the same as either 5v or 0v respectively. This output voltage will control the relay module that will be able to turn the module on or off. It also takes the input of one button on the face of the device where two short double presses trigger a manual override for turning the device on and off, and a long hold triggers the pairing sequence. Additionally, it will use a different GPIO to write the current state to an indicator LED on the face of the device. In order to do this, we ideally want a microcontroller that has a fast clock speed, has the ability to enter low power modes, uses a low amount of energy, has several GPIO pins, has pin interrupt functionality, and has enough internal memory to store all of our code. Most importantly, it should have a relatively low cost if possible.

3.1.1 Microcontroller Options

There are many different microcontrollers available on the market, which means that there are a lot of options to choose from. Each of these microcontrollers have metrics that differ between other microcontrollers, where each metric is based on the microcontroller’s intended use. For example, some are intended for performance-based projects and some are intended for low power usage to ensure a long battery life. The best way to choose the correct microcontroller for a project is to consider what these different metrics are, then compare them to see which microcontroller fits the needs of a project the best. That is exactly what we do in the following sections. We first elaborate on the details of several microcontrollers from a reasonable set of ones we could potentially pick from. After that, we put the metrics into a table for easy comparison, then discuss the metrics and their importance to our project. Lastly, we make a decision based on the conclusions we drew from the previous sections. For our project, we decided on the ESP32 microcontroller since it meets many of the requirements mentioned above, which is further explained in the below sections.

3.1.1.1 Arduino Atmel Microcontrollers

There are several microcontrollers commonly called “Arduinos” that are used by developers all over the world. They have an extremely well-documented set of resources and a friendly IDE (Integrated Development Environment) that makes working with them remarkably easy and straightforward. They mostly run at lower clock speeds since they are typically used for hobbyist projects. They also use very little power. However, the most popular models do not have built-in wifi or bluetooth. Additionally, they are open source, allowing other manufacturers to make exact copies for very low prices. Below are some examples of some of the most popular models used.

3.1.1.1.1 ATMega328P

This model is the most popular for hobbyists and developers alike, commonly found in an Arduino Uno development board. It can be purchased at a price of $1.88 per unit making it remarkably cheap. As mentioned before, it has an extremely large amount of documentation, examples, and developer projects that use it. This is a result of it being so popular, which makes it a very reliable microcontroller to use.

This microcontroller operates in the range of 1.8V to 5.0V and up to 16 MHz in clock frequency. It has 32KB of non-volatile memory and 23 usable GPIO pins as well, where 8 of which have ADC functionality. It can also be put into sleep modes to reduce power as well. Despite all of these great features, the microcontroller is relatively large, needs support components, and does not have built-in bluetooth. This means that we would need to use an external bluetooth module, like the HC-05 for example. The biggest constraint of our project is space, so using a bigger microcontroller with an external bluetooth module is not very optimal for us.

3.1.1.1.2 ATMega2560

This model is another very popular model made by Atmel. This is like the “older brother” of the Arduino Uno. It has a unit cost of as low as $1.55 per unit, and a smaller form factor than that of the ATMega328P. It operates in the range of 1.5V to 5.0V and up to 16MHz in clock speed. It has 4096 KB of non-volatile memory and 54 GPIO pins. It can also be put into sleep modes. The most impressive metrics about this microcontroller is the number of pins and the size of its memory. Compared to other microcontrollers, this has a much larger number of pins, making it able to manage more than most microcontrollers. Additionally, it has a large amount of program memory, allowing for its programs to be much more lengthy and complex. Though the microcontroller has a smaller form factor, it still suffers from not having any built-in bluetooth capabilities. This would result in needing an additional module, which would occupy too much space in our project.

3.1.1.2 TI MSP430 Microcontrollers

These microcontrollers are based on the idea of bringing an easy microcontroller platform to developers in order to make complex projects. The name MSP430 is actually a name for a family of microcontrollers, meaning that there are several variants under this umbrella term that each focus their resources on some specific need. These may be designed to meet the needs for clock speed specifications, power consumption specifications, or a healthy share of each for general purpose use.

3.1.1.2.1 TI MSP430G2553

This model is fairly popular among developers using Texas Instruments hardware since it caters to general use. It has a cost of around $3.45 per unit and an average form factor that is neither too large nor too small. It operates in the range of 1.8V to 3.6V and up to 16MHz in clock speed. It has 16KB of non-volatile memory and around 24 usable GPIO pins. It can also be put into lower power modes. This microcontroller is great for general purpose usage, but does not have any wireless capabilities, making it a poor choice for our project.

3.1.1.2.2 TI MSP430FR6989

This microcontroller is also popular and is like the “older brother” to the MSP430G2553 microcontroller. It has a cost of around $9.34 per unit and a relatively small form factor since it comes in a surface mount package. It operates in the range of 1.8V to 3.6V and up to 16MHz in clock speed. It has 128KB of non-volatile memory and around 40 usable GPIO pins. It can be put into low power modes as well. One of the interesting features about this microcontroller is its built-in RTC (Real Time Clock), which can accurately measure time. However, similar to the arduino microcontrollers, this does not have any wireless features, which makes it a poor choice for an IoT (Internet of Things) project. Additionally, the unit cost is considerably higher, which makes it a less desirable choice.

3.1.1.3 STM Microcontrollers

The STM brand of microcontrollers are somewhat popular among developers, but definitely not as much as the Arduino or Expressif microcontrollers. They are a decent alternative to either though. They are relatively cheap and have a fair amount of documentation and resources to help with development.

3.1.1.3.1 STM32F103C8T6

This model of microcontroller is an average microcontroller, targeting general purpose use. It operates in the range of 2.0V to 3.6V and up to 16MHz in clock speed. It has 128KB of non-volatile memory and around 37 accessible GPIO pins. Additionally, the unit cost is around $7.00. This microcontroller faces the same issue as the Arduino and Texas Instruments microcontrollers as well, being that it does not have native wireless communication support. This means that, in order for it to work for our project, we would need an external wireless module. When considering the size constraint we are dealing with, adding extra elements is not ideal. As a result, this microcontroller is not very ideal for this type of project.

3.1.1.3.2 STM32WB55

This microcontroller is not as popular as some of the more well-known ones on the market. However, it has a lot of functionality packed into one package. It has a unit cost of around $8.47 and a relatively small form factor. It operates in the range of 1.7V to 3.6V and up to 32MHz in clock speed. It has 32KB of non-volatile memory and around 30 accessible GPIO pins. It also has the ability to dynamically track the power it uses and adjust its resource use accordingly. This is probably the most unique feature this microcontroller has. This was under decent consideration for our project, but it has very little documentation and examples about how to use it easily, which is a large downside for us. I imagine that if fully necessary, this would be a fantastic microcontroller for a mobile, battery-operated IoT project.

3.1.1.4 Expressif Microcontrollers

These microcontrollers essentially target a specific market among developers; specifically, the IoT market. These microcontrollers focus on being low-power, relatively small, easy to integrate into projects, and having a lot of impressive functionality packed into a small space. There is one additional feature that surrounds these microcontrollers that is on par with Arduino microcontrollers, and that is documentation and resources. There is a large community of developers that share their projects, ideas, and solutions regarding the Expressif microcontrollers, which makes developing with them much easier. Additionally, the Arduino IDE supports programming these microcontrollers, which makes development that much easier.

3.1.1.4.1 ESP8266

This microcontroller was one of Expressif’s first main microcontrollers to hit the market, paving the way for easy IoT development. It caters to this because it has both Wi-Fi and bluetooth built into the package. It has a unit cost of around $1.60 and is packed into a decently small package. Typically, the microcontroller itself is fitted onto a small PCB that houses additional support components to make everything operate properly. This extra PCB assembly makes using the chip much easier, and is not very large either. It operates at 3.3V and typically at 80 MHz in clock speed. It can also be overclocked to up to 160MHz. It has 32KB of non-volatile program memory as well as 13 accessible GPIO pins. As you can see, this was a feature-packed microcontroller to release to the public, allowing for a relatively easy way to develop IoT devices.

3.1.1.4.2 ESP32

This microcontroller was Expressif’s next big microcontroller to be released. They took the opportunity to improve on some of the shortcomings from the ESP8266 and add extra features, which would become the ESP32. To say the least, the additional features they packed into this device is nothing short of amazing considering its price. It has a unit cost of around $1.95 and fits in a form factor hardly bigger than the ESP8266. It is also fitted on a PCB to make using it easier, just like its predecessor. It operates in the range of 2.6V to 3.6V and typically at 80 MHz in clock speed. It can also be overclocked to up to 240MHz. It has 512KB of non-volatile memory as well as 34 usable GPIO pins. Additionally, it has some unique features such as native capacitive-touch support, a cpu temperature sensor, native support for infrared and CAN bus communication, and several more. This microcontroller not only has a lot of functionality packed into a small space, but the price is remarkably cheap and there is a great set of community resources surrounding it.

3.1.2 Microcontroller Comparison

Below is a table to help compare each of the metrics discussed in the previous section. This way, comparing each of them should be much easier. For future reference, the option that we chose to use is highlighted in red.

Table 2. Microcontroller Comparison

	Cost	Memory	GPIO	Max Clock Frequency	Wireless Communication
ATMega328P	$1.88	32KB	23	16MHz	No
ATMega2560	$1.55	4096KB	54	16MHz	No
TI MSP430G2553	$3.45	16KB	24	16MHz	No
TI MSP430FR6989	$9.34	128KB	40	16MHz	No
STM32F103C8T6	$7.00	128KB	37	16MHz	No
STM32WB55	$8.74	32KB	30	32MHz	Yes
ESP8266	$1.6	32KB	13	160MHz	Yes
ESP32	$1.95	512KB	34	240MHz	Yes

When evaluating and comparing the different metrics that describe microcontrollers, the most important thing to do is to identify the metrics for each device that will be compared. In addition to the metrics that have been laid out in table 3.1.2, there are others that we decided to not use since they are less important. This includes interrupt latency, architecture, instruction execution rate, security, hardware interfaces, and temperature tolerance. These are metrics that our team is less concerned with, since our application is not so sensitive to them. For the ones that we are more concerned with, there is an analysis for each of them below.

3.1.2.1 Cost

Cost is one of the most important metrics that we want to review. If there is a perfect microcontroller with each metric being in the range of parameters that we need, but it is just too expensive, it would not be realistic to buy that specific microcontroller. Additionally, since each member of our team is a college student, we want to keep costs as reasonable as possible. When looking at microcontroller costs, it is important to consider that the cost you find is for the microcontroller itself, not accounting for any additional support components needed to operate the microcontroller. Regarding the options listed in the above figure, it is clear to see that there is some variation in the cost. Some devices are placed in the lower range of $1 to $2 per unit, whereas some of them can be placed in the more expensive range of $7 to $9 per unit. Some of the models that are lower in cost are the Atmel Microcontrollers and the Expressif Microcontrollers. As a result, these microcontrollers are good potential candidates for being used in our project.

3.1.2.2 Memory

Memory is an important metric to discuss, since it directly correlates with the complexity of the program you can store on your device. Some devices may need more memory to store more instructions, which are the fundamental building blocks of a program. For example, some programs may need to render an image on a display, which usually requires a bit more complexity. This is especially true if a display is having animations run on it, since doing so requires a lot of computing and processing. Using this same example, it is easy to see that memory is also related to what you can store for what you want to render on the display. For example, you may want to store bitmaps locally in the memory for quick access. This is one of many examples of why memory can be so important.

However, not all programs need so much memory. For example, if you are just checking a sensor and sending data over bluetooth, your program is not very complex. This means that you may not need very much memory to store your program. Other than some additional features, this example greatly depicts how our project will fundamentally work. So in our case, we are not as restricted as we are with cost and can either use more memory or less memory as needed. If we did have more memory than we needed though, it would give us the opportunity to add more features in the future if we desired to do so. Regarding the options above, we could either use a microcontroller with memory in the range of 16KB - 32KB, or something with a bit more capability in the range of 128KB to 512KB. Though there is not really a restriction on the amount of memory we could have available to use, the ATMega2560 has 4096KB of memory, which is a bit too much than we would ever use.

3.1.2.3 General Purpose Input/Output

General Purpose Input/Output, or better known as GPIO, is also important to evaluate. These are the pins that are located on the outside of the microcontroller, which allow it to interface with the surrounding world. These pins are used to either read a digital input signal, or generate an output signal. This functionality is commonly used to turn an LED on/off or read the state of a button. Even if their use is simple, they still allow for interacting with the world. However, they can also be used to do more complex operations, such as PWM, communication protocols, and reading changing voltage. These operations allow it to communicate with other sensors and systems to expand its functionality.

Since GPIO pins are like a connective tissue for the microcontroller to the outside world, it makes sense to say that the number of them can greatly limit the microcontroller's functionality. More of them means that you can connect more sensors, buttons, and other inputs. Less of them means that you become much more limited. For our project, we only need to connect a few pins to the outside world. We need to be able to read the state of input buttons, read the current sensor, turn on/off an indicator LED, and turn on/off a relay to control the outlet's flow of power. That being said, we have a lot of flexibility since we do not need many pins to fulfill our hardware requirements. Regarding the options above, there is a large range of pins. Realistically, a pin count of approximately 30 should be sufficient for our needs. It is possible to use fewer pins, but based on the data above, that number of pins is an estimate for the average, easily available microcontroller.

3.1.2.4 Clock Frequency

Other than cost, clock frequency is arguably one of the more important metrics that must be evaluated when deciding on the right microcontroller for a project. The clock inside a microcontroller is the fundamental method for timing operations. The clock is used in a variety of ways, such as making sure instructions are executed in the proper order by proper timing or to drive PWM on the output of a pin. It can also be used to count numbers over time, which is perfect for timing events that you may want to trigger with your program. The more valuable observation to make about the clock is that a faster clock allows you to have a shorter response time. In terms of a sensor, that means that you can have a higher accuracy since you can take more sensor samples per second. In our case, a higher clock is preferable since we want to have a high accuracy. Regarding the options above, there are only a few different clock frequencies. For general purpose use, most microcontrollers run between 16MHz to 32MHz in clock speed. Some other ones with more power behind them can run at faster clock frequencies between 160MHz and 240MHz. In our case, using a microcontroller with a higher clock frequency is preferable, so the Expressif microcontrollers would be best in terms of this metric.

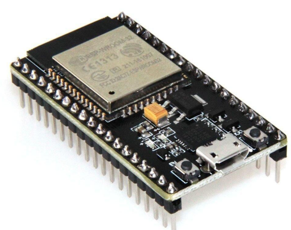
3.1.2.5 Wireless Communication

Wireless communication is not a commonly analyzed metric when looking at microcontrollers. In most cases with microcontrollers, these are often combined into one package that provides both. In our case though, where we are developing a smart IoT device, this metric is rather important to consider. Having wireless capabilities means that a microcontroller has access to hardware that can communicate with other devices using either bluetooth or Wi-Fi protocols. When thinking about wireless technologies with respect to microcontrollers, there are two main options to choose from.

Microcontrollers can either have wireless capabilities built in, or need an external device to implement them, which directly relates to the size constraint of the project. Projects with more size leniency can afford to have this extra hardware, but projects with less size leniency are significantly more restricted. In our case, our project has a rather restrictive size constraint, which means that adding extra hardware for wireless capabilities would be remarkably difficult. That being said, a microcontroller with wireless capabilities built into the device is much more preferable, since it saves a lot of space. Regarding the choices above, the Expressif microcontrollers are a better option since they have wireless capabilities built into the device.

3.1.3 Microcontroller Decision: ESP32

Figure 6. ESP32



For this project, we have decided to utilize the ESP32 Microcontroller. The reason for this choice is due to the fact that its different metrics strongly align with the needs of our project. These include its low cost, large memory size, sufficient GPIO pins, high maximum operating frequency, and integrated wireless capabilities.

The integrated wireless capability is one of the more important metrics that helped us decide on this microcontroller. Since our project has a limited amount of space, having a microcontroller with built in wireless hardware was practically essential. Apart from that, it can be put into low power modes, allowing it to only use the smallest amount of energy it needs to operate properly. One of our project’s main goals is to help save people money from their electric bill, so having the processor itself use a small amount of power will help meet that goal as well.

The low cost is the next most important metric that factored into our decision. The cost of the microcontroller itself is only $1.95 per unit, making it a great choice for this project. As a result of that low unit cost, the development kits using this microcontroller have a low cost as well. Similarly, this microcontroller has a rather high maximum clock frequency. This will allow us to have a high accuracy with our sensors, making the system more accurate for the user. Though this is the maximum frequency, the microcontroller can be set to different frequencies as well. This adjustability can allow us to set the microcontroller to the best frequency for optimal accuracy and power usage.

Additionally, this microcontroller has a large memory size, which will allow us to store our main code and have room for additional features later on. This is important since it gives us the freedom to write whatever code we need. This microcontroller also has sufficient GPIO pins, which means that we will have as many as we need to fulfill the device’s hardware user interface. It also means that we should have several unused pins, which can allow us to add additional features later on if needed.

After elaborating on each of the different metrics above, it’s clear to conclude that the ESP32 is the right choice of microcontroller for our project.

3.2 Interface Selection

For our smart outlet, there are a few different options we can use to display the output. Other smart plugs on the market use an LED/LCD display to show their results. There are some problems using a display such as that with our design. It will make the outlet very clunky and it could possibly not fit in the wall. Size is already a pretty big constraint on our product so that is a strike for having a display. Also, with that design, the outlet would look rather ugly which potential buyers would not like. Finally, the power usage of the smart outlet itself would increase which could potentially lessen the power going to connected devices and in fact increase your total power bill which is what we are trying to in turn, decrease. The average smart plug uses about 1-2Watts of power. Since our product is going to be the whole outlet, having a display on there would increase the amount of power that will already be increased due to it being the whole outlet itself and not just a single plug. However, there may be some benefits to using some sort of screen and therefore we will compare the most common technologies available to us.

3.2.1 OLED Screen

If we were to utilize a screen, one of the options available to us would be OLED. OLED stands for organic light emitting diode. It utilizes a thin carbon based film built into the screen. The way OLED works is that each individual pixel in an oled panel provides its own illumination. Therefore, there is no need for any backlight when compared to other screens such as LCD. Furthemore, OLEDs can be significantly better in power efficiency than LCD when using phosphorescence. OLED is known for the vivid display colors and viewing angles but in our case we would not focus on that aspect. The OLED screen we would be looking at would be monochrome. Monochrome OLEDs have a higher contrast than monochrome LCDs so it would be better if visibility was an important aspect of our device.

Other aspects that can be of importance would be temperature variability and sunlight readability. OLED is consistent at most livable operating temperatures from negative forty to positive eight-five and due to the monochrome contrast would have great visibility in sunlight. Furthermore, OLEDs are able to refresh in microseconds which is very fast. This leads to increased readability and visibility overall in regards to the whole package. As a last note, OLED has a wow factor in that it can be incorporated into a multitude of projects as there are a wide variety of screens available. Such as transparent screens and flexible screens. If selected, our device would come with a transparent OLED screen as seeing the back of the wall begin the outlet could bring peace of mind to the user as nothing could be hiding there.

Of course OLED panels come at a higher cost than when compared to LCD screens. If we needed the increased visibility and readability when compared to other screens, then OLED would be a sure winner but the higher cost especially when considering a larger manufacturing plan would be larger than using other displays.

3.2.2 LCD Screen

LCD stands for liquid crystal display. They are commonly found in smaller applications and are seen as the common go to for other energy monitoring devices currently on the market. LCD screens work by utilizing liquid crystals that are embedded into the display screen to display an image. Within the screen there are several layers of materials such as a polarized filter and an electrode. LCD screens are so common that the most popular watch in the world utilizes an LCD screen. LCD screens do not natively have a backlight when compared to OLED screens and therefore can use significantly less power when no backlight is required. LCD displays typically have a longer life than OLED as they do not degrade as much over time.

In regards to visibility and temperature, LCDs work well in day-to-day temperatures but become dimmer or low temperatures and slightly fade at high temperatures. For most applications with an outlet this would not affect standard usage but it is something to take into account in regards to extremes. Furthermore, in bright sunlight, LCDs can become more difficult to read and this becomes worse if there is no backlight. In darkness the LCD becomes unreadable without a backlight which can be annoying in some applications. LCDs are significantly slower than OLEDs as they refresh in the milliseconds range as opposed to the microseconds. This leads to less readability if we are to compare the quickness of OLED panels.

Overall the biggest advantage is that the LCD is less expensive than the OLED especially when not having to worry about a backlight. LCD is a simpler display and therefore has limitations in the fact that it is not as innovative as E-ink or OLED but it is reliable and affordable.

3.2.3 E-Ink

E Ink or electronic ink is a screen technology that mimics the appearance of regular ink on paper. Furthermore, they reflect light rather than emit it when compared to other types of panels like OLED and LCD. These displays are good for low power projects and can maintain the text or image being displayed for a long time. This is due to the fact that it places an active material or ink within an encapsulation and uses electronics to activate it. The larger the screen when compared to LCD or OLED the more saving in energy. For our case since we are using smaller screens the energy saving might not be the benefit we need.For visibility, since it reflects light and acts like a paper, it requires light to be read and will not be readable in the dark.

E-ink screens have other aspects that can lend itself to more ruggedness when compared to LCD and OLED. Due to its construction, E-ink displays can be lighter than comparable screens at the same size. Furthermore, the screens can have special applications such as being flexible. A negative of e-ink can be their refresh rates as they are mostly used for images that will be on the screen for a substantial amount of time and are not expected to switch rapidly. Most e-ink displays have a refresh rate nearing somewhere in the hundreds of milliseconds which can be annoying but not a deal breaker when dealing with measuring current. Also, e-ink displays do not have the economies of scale when compared with OLED and LCD. That is to say they are more expensive as there are significantly less options when compared to LCD and OLED.

3.2.4 Phone Application

One option that we have not touched upon yet is a screen in which we do not have to manufacture. By making the user use their own electronic device for communication of information, we allow the physical hardware time an easier time when working with the actual device. This is especially important when considering the limited amount of space available to to create something within the confinements of empty wall space. However this shifts effort from physical to software but that may be easier. Also this adds more complexity in regards to communication methods as that becomes an even more important aspect than before. The app would connect to the outlets via bluetooth and wifi. It is easy nowadays to find a board that has bluetooth and wifi capabilities so it is not like we are going on a wild goose hunt to achieve the connectivity we want.

Having the results on an app makes it much easier and convenient for the user. The output can be read from the app even if the user is not at home. That way, if the user wishes to turn the outlet off remotely or with a timer, they have the ability to do so. The app will be easier to read since one would not need to bend over or get on their knees to read what the outlet is showing. The app also provides another level of customization to set certain outlets to turn off at certain times of the day or what-not. The possibilities are numerous. The app would also be able to tell someone their power history. It could easily keep track of how much power was used on the outlet a week ago. That could enable the user to realize they forgot to unplug their hairdryer and not to do it again because it cost them a lot of money. The app also helps the outlet look simple, refined, and without clutter or clunk. Lastly, an app is free to create whereas a display attached to an outlet costs more money.

Some of the negatives in regards to utilizing a phone application instead of a screen would be connectivity. If there are any issues with the data communication method we are using then you will not be able to get any data. This can be mitigated by having the microcontroller be preset with some sort of limit to stop if necessary but can be annoying to work around with. Even in these modern times losing connection can be very annoying. Another aspect is that we are shifting the burden from us to the user. If the app experience is not perfect then the user will be annoyed and feel negative emotions to the product.

3.2.5 Screen Selection

When compared to the options above. Overall, having it on an app gives the outlet a much cleaner look, is much easier for the user to control, and is more cost effective for the team to create. It does shift some issues to the user but it works out for the long goal. If we were to select a screen however, I would select an OLED screen as it is extremely versatile and interesting to work with when compared to LCD's despite its higher cost

Table 3. Panel Type Comparison

	Advantages	**Disadvantages**
OLED	Readability, Refresh Rate, Innovation	Cost
LCD	Reliable, Cost Effective	Readability without backlight

E-Ink	Low Power Consumption, Innovation	Refresh Rate, Cost
Phone Application	Cost, Versatility, Hardware Design	Wireless communication adds points of failure, Possible user experience

3.2.6 Smart Outlet vs Smart Plug with LED/LCD

We have been comparing how the function of these two different applications of the devices are but we have not seen how they would look in person. Below are two examples of normal outlets in their respective screen type. It is objectively clear that the outlet without the screen is smaller and more minimal in nature. The screen of the LED outlet takes up two spots of plug real estate in the home which can be a privilege in small dwellings. Either way the two plugs are both options in the market and the consumer is allowed to pick whichever they may want. Keep in mind that everything that is done on the screen for the device with the display would be able to be done with the app.

Figure 7. Smart Outlet vs Smart Plug with LED/LCD

3.3 Buttons and Switches

When working with a device that will most likely have some sort of wireless communication method through an application of sorts, it becomes convenient to have some sort of physical method of communication that the user can do if necessary. This case comes up when there are issues regarding wireless communication such as the signal being spotty around where the outlet would be or the router going down in the house. In these cases it becomes important for the user to be able to press a button or a switch to turn on or off the device.

3.3.1 Momentary Buttons or Push Buttons

A push button is a simple button used to turn the control circuit on and off. In this case this might be all that we need as we are working with a relay and a microcontroller. If we were not working with those we would need a button that would be able to handle any of the current flowing and would also have to be latching. A simple push button is usually thought to be a momentary button as it only works when the button is actuated. This would not work with our device as we would like the outlet to remain on and off for an extended period of time. Not only do one of those while the button is being pressed. However, since we are using a microcontroller, we should be able to read changes in the circuit regarding whether or not it was closed through the actuation of the button and use that for our controlling of the relay.

Overall, push buttons tend to be cheaper than maintained switches in applications as they can be used with microcontrollers to limit the rating that they need to be. They can be used as on and off control but do not necessarily need to be a path of current. Other types of momentary switches are joysticks that only actuate when pushed against sideways and reed switches.

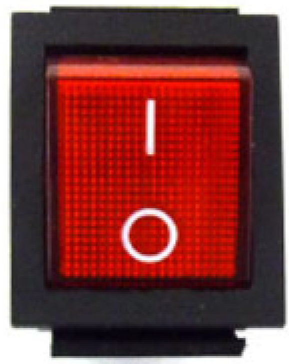
3.3.2 Maintained Switches or Latching Buttons

Maintained switches or as they are commonly known toggle switches, are used to maintain the state of a system until you change the position of the switch to the other actuation. If not using a relay this would be an important part of our system as we would not need to use the microcontroller and software in order to make up for the shortcoming of a momentary button. In the case of our outlet, the user would be able to set it to on and off without caring if the esp32 was plugged in or not. However, this maintained switch would be used as a primary on and off control and would therefore be directly in the path of current. This means that we would need to make sure that the maximum amount of current would not be exceeded and would want failsafes to prevent any sort of accident from occurring, Considering that this is an outlet with a possibility of valuables being connected to it or it being near a restroom, this may not be the best option.

Another issue is that this takes away from the whole design aesthetic that we are trying to create with our device. Toggle switches change the sleek package a device may come in as they are usually bigger or when they are toggled on, are a different height than when toggled off. We would also need to use a higher quality one in order to make sure that it could handle higher current if a device was requesting it. By higher current I mean around 3 amps. These switches would cost around 5 dollars per switch and when tracking account the economies of scale this would be a significant amount of money than a cheaper button that you don't have to worry about.

3.3.3 Button Selection

Figure 8. Push Button vs Switch Button



Overall our other components allow us to choose the more affordable option of a simple push button as we are using a relay and a microcontroller. If we didn't have those then maybe a higher quality switch would have been warranted as it has the benefit of always being in a desired state. If you want it on you know its going to be on and if you want it off you know it's going to be off. That is an aspect we are sacrificing with our push button.

Table 4. Button Comparison

	Advantages	Disadvantages
Push Button	Budget-friendly, affordable	Forced to constantly actuate for changes
Toggle Switch	Always desired state	More cost, Complexity increases due to current rating

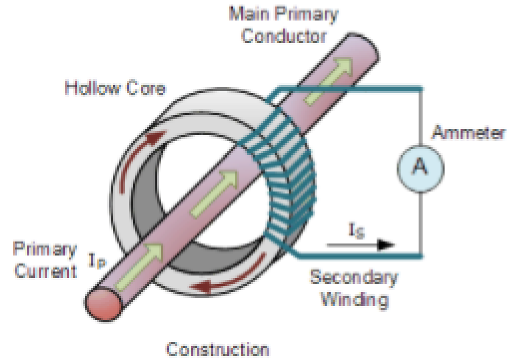
3.4 Current Sensor

After doing much research, we came to the conclusion of using potentially two different types of current sensors. An AC current transformer sensor or the ACS712 current sensor. Below discusses our thoughts on both and why we chose one over the other.

3.4.1 AC Current Transformer Sensor

Current Transformers are sensors that measure alternating current. We are going to utilize a current transformer in order to measure the electricity consumption within the connected device. A current transformer measures an input current and produces an alternating current proportional to the primary current. For example we can attempt to read ten amps from the primary current but utilize the proportional current of a couple hundred of milliamps. The value of this is that we can utilize the transformer to read the real time current flowing through a given circuit. The way a current transformer works is through three main components: a primary winding, a magnetic core, and a secondary winding. This is shown in the figure below.

Figure 9. A/C Current Transformer



The primary winding in a current transformer consists of very few turns. It can be a single flat turn, a coil of heavy duty wire wrapped around the core, or even just a conductor placed through a central hole. The alternating current flowing through the primary produces a magnetic field in the core. This induces a current in the secondary winding circuit.

The second winding has much more coil turns on a laminated core of low-loss magnetic material. This core has a large cross-sectional area so that the magnetic flux density created is low using much smaller cross-sectional area wire. This is done as the current must be stepped down as it outputs a current independent of the load. From here the current will either supply a current into a resistive load or a short circuit.This burden resistor completes the current transformer secondary circuit and its value is chosen to provide a voltage proportional to the secondary current.

There are a few types of current transformers such as: wound current transformer, Toroidal Current Transformer, and Bar-Type current Transformer.

In wound type current the primary and secondary winding are both present. The primary winding needs to be physically connected in series to the current carrying conductor.

Bar-Type current Transformers contain a copper or aluminum busbar surrounded by the secondary winding wound over the ferromagnetic core. The busbar acts as a single turn primary winding. They are directly connected to the current-carrying conductor. They are also known as bar-primary CTs.

The one we are focusing on is a Toroidal Current transformer. Toroidal transformers are power transformers with a toroidal core on which the primary and secondary coils are wound. When a current flows through the primary, it induces an electromotive force (EMF) and then a current in the secondary winding, thereby transferring power from the primary coil to the secondary coil.The unique shape of the toroidal transformer allows for shorter coils, reducing resistive losses or winding losses and improving overall efficiency.

Toroidal Current Transformers also have a compact size which makes them ideal for our use regarding the smart Outlet. One small benefit too is that this type of transformer makes less noise when compared to others. This is a good fact as this device may be used in locations where noise levels matter such as a bedroom.

This transformer can have a split core which allows us to open and install it where we need to in order to easily measure the circuit of which we are measuring. Some transformers have built in safety standards such as zener diodes that will break down and not allow it to be short circuited. Our Noyito Split-core current transformer will allow up to 100A in the primary core with an output of 50mA.

We were originally going to use a current transformer sensor but ultimately decided on an alternative.

3.4.2 Current Sensor

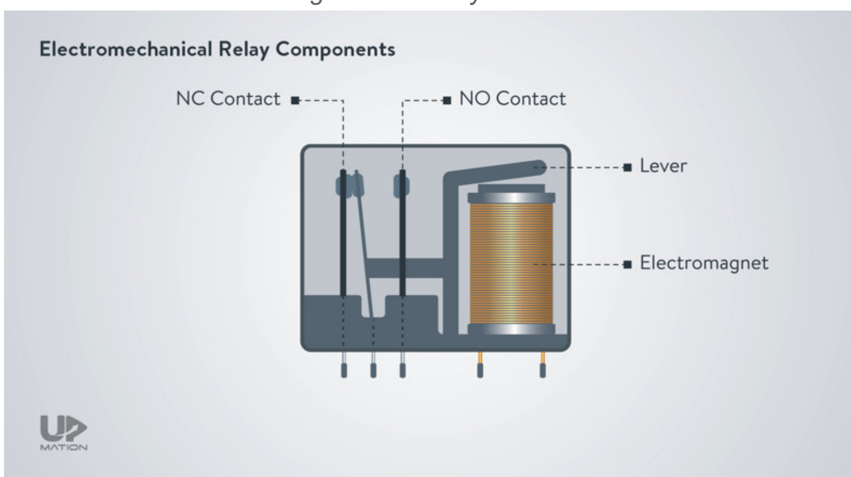
The smart outlet will make use of the ACS712. This is perhaps one of the most important components of this project. This current sensor has a 80KHz bandwidth, 66 to 185 mV/A output sensitivity, and a total output error of 1.5% at TA=25°C.The ACS712 works by using its hall effect sensor to read the current coming through by the use of its magnetic field generation. Then, the hall effect sensor creates a voltage that is proportional to its magnetic field that then measures the current. Originally our project was going to use a current transformer sensor but due to size constraints, the ACS712 worked out better.

3.5 Relay Module

A relay module is an electrically operated switch that can be turned on or off by an electromagnet to allow flow through the circuit or not. They are made to be compatible with various low source microcontrollers such as the ESP32 that we are using. One of these modules usually consists of a solenoid, a moveable iron structure and at least one set of contacts. The moveable structure is linked to at least one of the contacts. Also, a spring keeps the structure in place. When an electrical current goes through the coil, it causes a magnetic field that moves the structure. When the structure moves either way, it makes or breaks the connection. That in turn, powers on or off the circuit for the device to function.

We need our relay module to be able to withstand at least 70 amps for peace of mind. We would not want the outlet to blow while using it so durability on this part is a must. If it breaks, the device may be stuck in the on or off position which would be a big problem for our product. One of the key features of our smart outlet is that someone is able to remotely or physically cut power to the outlet to save power which in turn saves money for the user. There are a few other types of relay modules other than the electromechanical relay that is the most common and was previously described. Within an electromagnetic relay there are different ones that latch or do not latch and have a single or a double throw. This means that the latching type has one coil that remains in a constant position. The non latching does not have a coil that stays in a constant position. A single versus a double throw differs by the number of terminals. The single throw is a simple switch that chooses between on and off. A double throw can switch between the two terminals to provide power to one circuit or the other which can provide selecting power to one or the other. Other two types of relay modules are solid state relays and reed relays.

Figure 10. Relay Module



3.5.1 Solid State Relay

A solid state relay or SSR uses semiconductors to switch the conduction and disconnection of high voltage loads. Usually, SSR contains an LED and a photosensitive MOSFET. When current flows through the circuit, the LED flashes and the MOSFET picks that up and tells the triode for alternating current or the silicon controlled rectifier to switch the load and turn on the device. Due to these components, an SSR is really quiet and efficient but extremely expensive.

3.5.2 Reed Relay

A reed relay is a switch with magnetic strips which are called reeds which is where reed relays derive its name from. The switch and reeds are contained in a glass tube that has inert gas that protects it from corrosion. These parts move due to an external magnetic field or an induced magnetic field. The magnetic field moves the tube so that way each reed on each side of the tube can touch which creates the flow of the circuit. This allows the circuit to function without the moveable iron structure that electromechanical relays require. Due to this, reed relays require low power consumption but are susceptible to inductive loads which can easily make them blow if some sort of protection is not made for it.

3.5.3 Polarized Relay

Polarized Relays are sensitive to the direction of current by which it is energized. It is like an electromagnetic relay but it uses DC voltage and a constant magnetic field to move the armature of the relay back and forth. This kind of relay uses permanent magnets, electromagnets, and the armature to build its circuit. The way it works is that it uses magnetic forces to move the armature back and forth instead of using springs like other relays do. To use magnetic forces to move the armature, this means that the armature itself must be a magnet to be able to be repealed to and fro. The armature is positioned in between two poles formed by an electromagnet. The electromagnet fabricates a magnetic flux when current is provided to it. When the electromagnetic force is high enough, it moves the armature into the second position.

When the current is interrupted, the force is not strong enough to keep the armature in the second position and the armature returns to the original position in position one. However, when there is no current in the system, the armature will remain in the position it is currently in. This is due to the fact that neutral is not stable in magnetic systems and the magnetic flux of the system keeps it in place. There are two very common polarized relays called differential and bridge. For differential polarized relays, the difference of the two fluxes of the armature magnet is where all the action is. For bridge polarized relays, the coil is instead divided into two fluxes and is what acts in the system. For the bridge type, the armature magnet does not split into two fluxes and remains one. The differential type is the most commonly used for practical applications. For our outlet, polarized relays seem to be on the higher end in terms of cost and we do not see a necessity in having it in our device.

3.5.4 Buchholz Relay

Buchholz relays either employ gas or use actuated relays. These relays are used to find very minor faults. They are most commonly used for transformer protection and are usually located in between the transformer tank and the conservator. Buchholz relays are solely used for oil immersed relays that are for power transmission and distribution systems. When an error starts to occur in the system, the oil level in the relay falls as gas begins to accumulate. This makes the float start to tilt and the mercury contacts are shut. The mercury contacts are the alarm of the circuit and this lets the mechanic know that there is a small error in the transformer. If a big error ever occurs, oil is rushed towards the conductor and this causes the mercury contacts to shut again and the transformer is then disconnected from the circuit. This relay has a very specific application and it does not pertain to our project, thus it will not be considered in use for our smart outlet.

3.5.5 Overload Protection Relay

Overload protection relays are partially explained in the name. They are designed to protect systems from overloading via current. There are a few different types such as bimetallic strip, electronic, and interchangeable heater bimetallic. This type of relay senses overcurrent by using heat operation. It heats up a bimetallic strip and releases a spring that operates auxiliary contacts that are in line with the coil. The coil loses energy when it senses excess current in the load. Then the system can be shut off. Although this is very useful in many applications, this is also not very useful in our smart outlet. Thus, we will not be using it.

3.5.6 Inverse Definite Minimum Time Relays

Inverse definite minimum time relays, or IDMT relays, are good for giving statistics on a circuit. They give definite-time values in relation to current at high values of the error current and inverse-time values in relation to current at low values of the error current. IDMT relays are most often used for safeguarding distribution lines. These relays can also put limits for the current and time. As described, the operating time and the error current are inversely proportional. The relay components contain a magnet that gets saturated for the current a little bit higher than the pickup current. IDMT relays get their name because when the error current reaches infinity, the time does not approach zero. We will not need an IDMT relay in our circuit so we will not be using one.

3.5.7 Relay Decision

For all these reasons, electromechanical relays are the most cost-effective and stable but are slightly slower than other relay options. We will be going with an electromechanical relay because it is the best option for our project.

Table 5. Relay Comparison

	Pros	Cons
Electromechanical Relay	Cost-Effective, Reliable	Slow
Solid State Relay	Quiet and Fast	Extremely Expensive
Reed Relay	Low Power Consumption	Sensitive to Inductive Loads

Polarized Relay		Expensive, Sensitive to Current Direction
Buchholz Relay	Good for transformer Relays	Specific Applications
Overload Protection Relay	Protect system from current overload	Specific Applications
Inverse Definite Minimum Time Relays	Circuit statistics	Specific Applications

3.6 Analog to Digital Converter

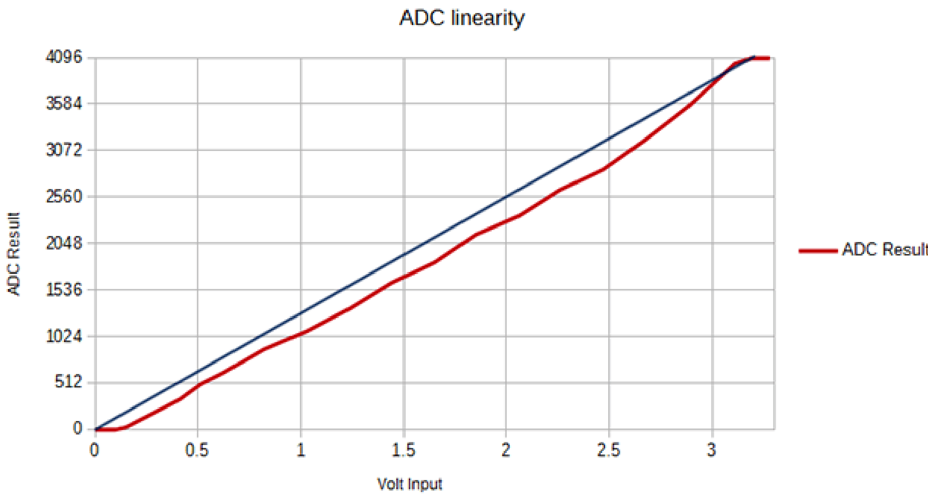
One of our group's goals for this project is to be able to accurately read the amount of power the outlet is using at any given instant. The way we intend on doing this is by reading the voltage across the sensor pins. We can then use that to calculate the power based on the relationships between the outlet current, sensor current, and sensor voltage. The most common way for a microcontroller to read analog voltages is by using an ADC, or Analog to Digital Controller. This works by taking in an input voltage and comparing it to a well-regulated reference voltage. Specifically, it goes through multiple comparators, which compare the input voltage to even smaller ranges of the reference voltage.

By doing these comparisons, it can narrow down the input voltage to a very close estimate and give back a binary number that corresponds to that estimated voltage. This provides a means to convert continuous-time and continuous-amplitude analog signals into discrete-time and discrete-amplitude digital signals. Essentially, the analog voltage can be turned into a binary number that can be used in calculations.

However, there is an issue with the ADC that comes built into the ESP32. As you approach the extremes of the voltage range defined by the reference voltage, the approximation becomes less and less accurate. If you ignore the extremes, the rest of the behavior is almost linear. This property of linearity is extremely important, since having it makes calculating numbers based on sensor data remarkably easier. As a result, we have decided to use an external ADC that is far more accurate and has more features, such as differential voltage measuring. In addition to linearity, there are other metrics that can measure an ADC's performance, which are further explained below.

In the figure below, you can see the behavior of the ESP32's ADC at normal and extreme values.

Figure 11. ESP32 Behavior: ADC Result vs Voltage Input



3.6.1 Analog to Digital Converter Options

Below is a list of options that we can use for our project, each with a small description and breakdown of their most important metrics. We want to compare the possible options we have and what would be better in terms of resolution and accuracy while also being realistic in price point for a device of our size. It would not make sense for the analog to digital converter to be extremely accurate if the cost was significantly greater than the rest of our device proportionally.

3.6.1.1 ADS1110

This ADC chip is a more simple chip that only has one input channel configured in a differential setup. It has a sampling rate of 128 SPS and a resolution of 16 bits. It uses I2C to communicate, which is a commonly supported interface type. This chip is not as well known as its older brother, the ADS1115, though it is still popular among embedded system and product developers. However, this chip is considerably slow compared to some of the other chips available, which makes it better suited for projects with a smaller precision requirement.

3.6.1.2 ADS1115

This ADC chip is a bit more robust than its younger brother, the ADS1110. It is very popular among developers and has a large amount of support. It has a sample rate of up to 860 SPS and a resolution of 16 bits. It also has 4 single-ended input channels, which can be configured as 2 differential input channels, and uses an I2C interface. This chip can also amplify the input signal if needed, making reading smaller signals easier. Additionally, it has breakout boards made by the company adafruit. Adafruit has a remarkable amount of support, examples, and tutorials that will make using the sensor that much easier. This will make the ADC component of the project that much easier to do, allowing for more time to work on the other parts of the project.

3.6.1.3 AD7705

This ADC chip is another popular ADC chip among developers with a fair amount of support behind it. It has a sample rate of ___ and a resolution of 16 bits. It also has 2 differential input channels, and uses a 3-wire SPI interface. Though this type of ADC is useful, it can be a little slower than what most developers need. For scenarios where you don't need to read an input very quickly, like a potentiometer for volume control, this type of chip is perfect to get the job done. However, for projects with more precise requirements, it is not the best option.

3.6.2 Analog to Digital Converter Comparison

Below is a table to help compare each of the metrics discussed in the previous section. This way, comparing each of them should be much easier.

Table 6. Analog to Digital Converter Comparison

	Sampling Rate (SPS)	Resolution (bits)	Differential Input Channels	Interface Options
ADS1110	128	16	1	I2C
ADS1115	860	16	2	I2C
AD7705	500	16	2	3-wire SPI

3.6.2.1 Sampling rate

Sampling rate is one of the two most important ADC metrics to analyze for our project. This refers to the number of samples that can be taken per second. In other words, it refers to the speed that samples can be accurately taken. It is clear to see how important this is, since it specifies the amount of time between samples, which when lower, gives us more data and makes the data set more complete. In general, it is desirable to have this number as high as we can get it, since that will allow us to capture more readings and have more data to analyze. Of the options above, the ADS1115 is the best candidate for this, since it has a rather large sample rate of 860 samples per second.

3.6.2.2 Resolution

This is the second most important ADC metric to analyze for our project. This specifies the accuracy of the reading compared to the actual value the ADC is trying to approximate. With a higher bit count, we can have more numbers to represent the readable voltage range, defined by the reference voltage. This means that the reading is more accurate, which is highly desirable for our project. With respect to the above options, they are all in equal standing since they all have a bit count of 16 bits. 16 bits is a high resolution, so despite the fact that all of the options have the same number, it will be highly accurate nonetheless.

3.6.2.3 Differential Input Channels

For our project to work the best, using a differential input channel is ideal. This means that the digital reading is the difference between the ADC value of two input pins connected to the chip. This is much easier, since we will not need to reference the sensor to ground. We can instead connect the sensor directly to the ADC module. Regarding each of the options above, they all have at least one differential input channel. As of right now, we only need one differential channel, so each of these is sufficient in terms of that. However, it may be beneficial to have an extra available in case we need it during the development of the project. For that reason, the ADS1110 is less preferable than the other options.

3.6.2.4 Interface Options

The interface option is the communication protocol that the ADC chip uses to communicate with the microcontroller. Based on the above options, we can use either 3-wire SPI or I2C. The microcontroller we are using, the ESP32, has both of these implemented in the chip with hardware to make using it simple and easy. As a result, we could communicate with any of the options above. SPI is faster than I2C, making it slightly more preferable. However, the ADC should have a cache, allowing it to sample as much as it needs to, preventing a communication protocol bottleneck. Essentially, any of the chips above could be used since we can communicate using either the SPI or I2C protocol.

3.6.3 Analog to Digital Converter Decision

For the final decision, we decided on using the ADS1115 chip. The primary deciding factors were its fast sampling rate of 860 samples per second, as well as having an extra differential channel. It also has a 16 bit resolution, which is higher than the ESP32's maximum 12 bit resolution. Additionally, it is a Delta-sigma analog to digital converter, which is known for improving amplitude axis resolution while reducing high-frequency quantization noise. The design of the Delta-sigma converters allows for more varied uses and a high resolution. They also over-sample the analog signal much more than the chosen sample rate, allowing them to take better averages and produce a more accurate reading. The use of multistage anti-aliasing filtering nearly eliminates false signals, reducing errors significantly. Since we want a high resolution ADC with high accuracy, this chip is ideal.

3.7 120V AC to 5V DC Converter

The smart outlet will need a form of power and therefore we will be using an AC/DC converter in order to provide power to the device. An AC/DC converter is what we will be using due to the fact that we will be powering on directly from the wall and must keep a steady flow of power. Of course there are standards we must follow and due to living in the United States the Alternative Current available to us is 120 Volts at 60 Hertz for Frequency. As the outlet will be connected directly to the wires we do not have to worry about what type of socket we will be using to connect the device. Outlets in the United States are type A or Type B.

The DC Voltage we are aiming for is 5 Volts due to the low power required to run the MCU and other components we are expecting to use.There are two main methods for utilizing an AC/DC converter. We can build the AC/DC converter or we can buy it. Both of these options have their own pros and cons.

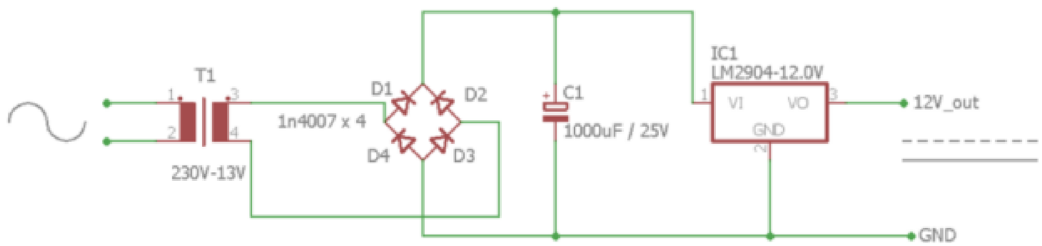
3.7.1 Building Half and Full Wave Rectifier

If we were to build the rectifier there could be a few options for the circuit we would utilize. In the simplest form we could use a single phase half wave controlled rectifier. Half wave rectifiers are mostly used in low power situations because of their significant drawbacks when compared to other options. Drawbacks such as the output being less than the input and wasting half of the power when negative ac cycle occurs. For these reasons the half wave rectifier is not the ideal situation for our device.

A full wave rectifier would be useful in producing a higher output than the half wave rectifier. Here it would also rectify the negative cycle component of the input voltage signal and therefore would not be wasting any of the negative cycles. This is the one we will be diving deep into.

This would be the first part of the circuit and the second would be to connect the transformer to a full bridge rectifier in order for the sinusoidal AC waves to have positive peaks. Therefore we would need to use a smoothing capacitor. The smoothing capacitor would be important to reduce the voltage drop and store the incoming energy. This smoothing capacitor would be the final step in order for us to have a DC output. Although this is the desired DC output, we would want to smooth out the variation in voltage with a voltage regulator. A representation of what could be with an AC/DC converter is shown below.

Figure 12. AC/DC Converter Full Bridge



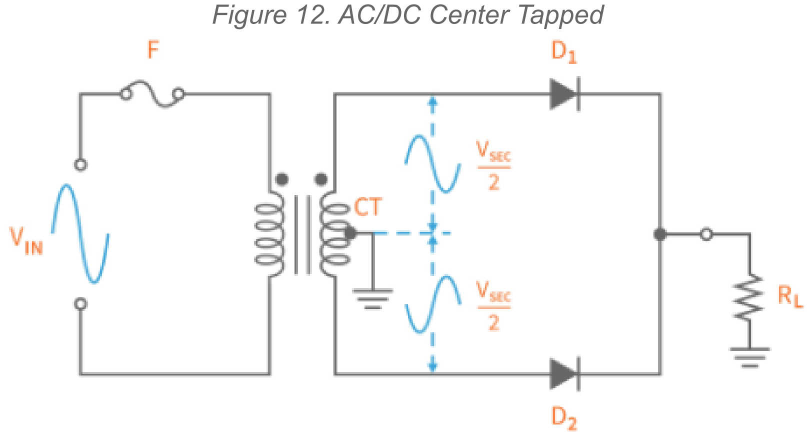
3.7.2 Building Center Tapped Rectifier

A bridge rectifier has the advantage of not using a transformer if no step-down voltage is required. Most of the time we will need to step down some sort of voltage into a usable range. One of the options would be a center tapped bridge rectifier. In this case there are only two diodes needed and therefore would be a little cheaper from the diode side. However we would need to take into account the fact that we would need to have a transformer for this circuit. Another benefit of working with a center tapped bridge rectifier is the fact that voltage drop across the center trapped bridge rectifier is half of that of the full bridge rectifier as it uses less diodes. This would help us in more critical measurements for voltage when compared to the full bride.

Building the AC/DC converter will require picking a transformer so that the AC voltage from the wall outlets can be converted. This would be the first part of the circuit and the second would be to connect the transformer to a full bridge rectifier in order for the sinusoidal AC waves to have positive peaks. Finally there would need to be a filter in order to remove the time in between these peaks. Therefore we would need to use a smoothing capacitor. The smoothing capacitor would be important to reduce the voltage drop and store the incoming energy. This smoothing capacitor would be the final step in order for us to have a DC output. Although this is the desired DC output, we would want to smooth out the variation in voltage with a voltage regulator. A representation of what could be with an AC/DC converter is shown below

.

Figure 12. AC/DC Center Tapped



3.7.3 Three Phase rectifier

Another rectifier that is common is the three phase rectifier. This sort of rectifier is more common in systems that require motor control or battery charging. These systems utilize higher power requirements in which a single phase circuit is unable to supply. This rectifier uses three diodes when it is a half wave 3 phase rectifier or 6 when full wave 3 phase rectifier, in order to half a low ripple factor in cases where it cannot be neglected. Overall 3 phase rectifiers have significantly lower ripple when compared to single phase rectifiers.

A negative regarding three phase systems is that they require 4-wire supply which is an extra complexity in which we would have to worry about.

Table 7. AC to DC Comparison

	Advantages	Disadvantages
Half Wave	Cost	Less output and more ripple
Full Wave Bridge	Simple, Cost Effective	Critical Data Measurements
Center Tapped	Data measurement	Specific Applications
Three Phase	Lower Ripple	Complexity, Specific Applications

3.7.4 Buying the AC/DC Converter

If we were not to build it and instead buy it. We must select the right AC/DC to work with our expected power draw. We will have to make sure the current is sufficiently high enough to power our device correctly. We do not have to worry about choosing a rating higher than what we need to use as it will work the way we want it to.

The AC/DC converter that we are entrusting our outlet with is an AC/DC converter module by a company called Delinx. The benefits of buying a ready-made module would be that it would be faster than buying all the components and working on them manually in order to create this by hand. Furthemore, since AC/DC converters have been in use for many decades and have been proven to withstand long stretches of time,this would probably be more consistent in regards to failure rates. Furthermore, in this ready-made module there are protections in place already to secure the safety of not only the device but the safety of humans who live in a home. The negatives of using a premade module would be that it is significantly more expensive than creating an AC/DC converter yourself. Furthermore it is larger due to the fact that it allows a wide variety of outputs in which we only require the use of 5 Volts. Other than that, there are probably not too many qualms about using this module as we believe the provision for it is valuable.

Table 8. AC to DC Converter Specifications

Rated Max Power	Variable
Input Voltage	70-277 VAC
DC Output Voltage	5 VDC
Item Weight	1.7 Ounces
Product Dimensions	37.5 x 9.5 x 15.3 mm

3.8 Wireless Communication Methods

In order for our device to be considered a smart device, it needs to contain some form of wireless communication in order to easily communicate between the user and itself. This will allow the device to upload data such as its power usage over time to some sort of database in which the user will be able to access and do actions based on those data points. In our case we expect the user to communicate with the outlet through some sort of app, but the means of which the app accesses the information can occur through many different protocols. We must compare and contrast the options available to us in order to see what communication method would work best for us.

3.8.1 Wifi

IEEE 802.11 is the standard of wireless LAN communications. The name behind this technology is what is known to us as Wi-Fi. This standard is overseen by the IEEE 802 LAN/MAN Standards Committee. Their main goal is to maintain the standards for networking and decide what is the best practices for local, metropolitan, and other area networks.

The two main technologies that are being used in today’s devices are 2.4 GHz and 5 GHz. These two frequencies have different use cases because there are advantages and disadvantages from both. The 2.4 GHz has a larger effective range because of the lower frequencies which means it can penetrate obstacles like walls. All W-Fi enabled devices will be able to use this lower frequency. This can also be a problem since multiple devices will be on the same frequencies, then that means that there is going to be more interference. While 5 GHz is faster in speed, the fact that it runs at a higher frequency means that it cannot go through walls as easily as the 2.4 GHz. This means that the access point has to be in a better location than the lower frequency one. If the user is not sure if the device is compatible with the higher frequency Wi-Fi, then the safer option is just to use the lower frequency for both range consideration and interference.

Wi-Fi channels are half-duplex, which means that two devices connected through a Wi-Fi channel can both communicate with each other, but not at the same time. If multiple networks attempt to use a channel simultaneously, the channel will be time-shared, meaning that the networks will take turns using the channel for an allotted amount of time. When two stations attempt to transmit simultaneously, that is known as a collision. In the event of a collision, the transmitted data becomes corrupted, and the stations are required to re-transmit. This reduces throughput, and in some instances, by a significant amount. To prevent such an event, carrier sense multiple access with collision avoidance (CSMA/CA) is used, which is a scheme that is used to handle how computers share a channel. It makes attempts to avoid collisions by having nodes begin transmission only when the channel is detected as “idle”. Once idle, nodes transmit all of their packet data at once. While there are still chances where collisions may occur, this does reduce the possibility by a good margin.

3.8.2 Bluetooth

Bluetooth is another standardized protocol for sending and receiving data via a 2.4 GHz wireless link. It is a secure protocol and is used for sending low power wireless transmissions between electronic devices.It was standardized as IEEE 802.15.1. The main goal for Bluetooth was to allow for short range wireless communication The Bluetooth Special Interest Group oversees the development of Bluetooth. Bluetooth works over ranges less than one hundred meters Bluetooth limits the number of devices that can connect at any one time. Bluetooth transmits radio signals by method of frequency-hopping spread spectrum (FHSS). With FHSS, Bluetooth divides transmitted data into packets and transmits each packet through a designated channel.

Normally, Bluetooth has 79 unique channels with each channel having a bandwidth of 1 MHz. It typically achieves 1600 hops per second while using a variation of FHSS called adaptive frequency-hopping (AFH). AFH is extremely effective at increasing resistance to radio frequency interferences by avoiding the crowded frequencies in the hopping sequence. This is great when dealing with static interference within specific frequencies within the range as all the channels afflicted with the said interference would be labeled as “bad” channels. This allows Bluetooth to avoid the bad channels in the hopping sequence. The instances when this wouldn’t be effective is if the frequency range was being afflicted by dynamic interferences in the system. If the “good” and “bad” channels within the frequency are constantly changing, then AFH won’t be effective at avoiding the interference in the system.

3.8.3 Zigbee

Zigbee is an IEEE 802.15.4-based specification for a suite of high-level communication protocols used to create personal area networks with small, low-power digital radios, such as for home automation, medical device data collection, and other low-power low-bandwidth needs, designed for small scale projects which need wireless connection. Hence, Zigbee is a low-power, low data rate, and close proximity wireless ad hoc network. Overall this means that it is designed for personal use and not business. As our device would most likely go into a home this is a possible area for Zigbee to shine.

A key component of the Zigbee protocol is the ability to support mesh networking. In a mesh network, nodes are interconnected with other nodes so that multiple pathways connect each node. Connections between nodes are optimized through a built-in mesh routing table. Zigbee is also less expensive than other networks intended for personal wireless.

3.8.4 Wireless Communication Selection

When compared to the other two common options we have come across, we believe that utilizing wifi will be a good mix of power and communication regarding our smart outlet. Furthermore, the ability to connect to the internet without being directly connected to a device will allow for some long range data transfer and communication between the user, the app, and the device. As a last note, the lower power consumption of wifi is an added benefit when considering the application of this device.

Table 9. Wireless Communication Methods Comparison

	Frequency	**Data Transfer**	**Range**	**Power**	**Connections**
Wifi	2.4 GHz	2.4 Gbps	30m	< 3 mA	Up to 7 Devices
Bluetooth	2.4 GHz & 5 GHz	3Mbps	100m+	Significant Power	Mesh
Zigbee	2.4	250 Kbps	30 m	340 uA in active run mode	Mesh

3.9 App Database

One of the main components of our app is the ability to store information with respect to different users. This can allow different users to store their information, preferences, and connected devices in a convenient way. By simply signing on, they are able to load all of that, allowing them to control their specific devices. In order to do that, we need to use a database. A database is an organized collection of information, usually controlled by a database management system. Based on the need to store data per user, this is exactly what we need for our project.

Databases can be stored locally on your machine or remotely on another machine. In this case, we want to use a remote database. This is because we want several user devices to be able to access the same set of data, which allows for logging into an account from any device. Generally speaking, these remote databases are hosted by certain sites, which morally have a web-interface. This web-interface allows an authorized person to see what data is in the database in order to manage it and make sure that both programs and apps are correctly interfacing with it. For an app to interface with a database, there is usually an application programming interface, or API for short. APIs provide a set of functions that an app can use to interact with other software. In this case, that is the database management system, hosted with the database itself. For example, if we want to add a new user to the database, we would use a function provided by the API that does that for us. If we wanted to update a password, we could use the API to do that too. The list of uses for an API is quite comprehensive, which is why they are so helpful.

When considering what database to use, there are several things to consider. One of them is how well your information is encrypted. When a device makes an API call, the message to the database can be intercepted and copied. This poses a vulnerability to your data, which is obviously not good. However, if the data being sent is encrypted, it makes interpreting intercepted data much harder, which improves security by a large factor. Another thing to consider is what features the API for the selected database provides you. For example, one feature can be the ability to keep a user logged in, allowing them to access a device without logging in every time.

Two additional things to consider are cost and consistency. Some databases allow you to set up a set of data for free, with certain restrictions. Others require you to pay in order to use their database systems. As far as consistency goes, some database systems are prone to work less accurately than other database systems. This usually shows up in the form of read-after-write errors. These are errors that arise from the idea that writing may take longer than reading, so it is theoretically possible to read the incorrect value after writing. For obvious reasons, reading the incorrect value when needed is a problem. The last most important thing to consider is reliability. This refers to how stable the database system is in terms of if it will be accessible online or not. Ideally, we want to use a reliable database system so we can always use it whenever needed.

3.9.1 Database Options

As mentioned in the previous section, there are several different databases we could potentially use. In the below section, we will introduce some of the more popular options and explain some of their metrics. This will greatly contribute to determining a final decision on which one we want to use.

3.9.1.1 Firebase

Firebase is a database system owned by Google, which has NoSQL. It is one of the most popular database systems available to developers at the moment, since it has a large amount of documentation and a lot of good reputation through Google. Firebase encrypts their data before it is sent to the database system from an API call, which helps to isolate user data. One of the nice things about the Firebase database is that it usually does a lot of the heavy lifting for you. For example, it has features built into the API that are directly related to user login systems. One of those features is the ability to check if users are logged into their account currently. This feature is particularly useful for keeping users logged in to their devices unless they explicitly log out. Regarding cost, the first GB used is free and the remaining data requires you to pay $25 for 2.5GB of data. It is also a very consistent system, allowing for data to be used whenever necessary at quick speeds. Regarding reliability, this service is very reliable and allows for users to have access to their data any time they need it.

3.9.1.2 Realm

Realm is another popular database system, which is made primarily for mobile app development. It was developed to help overcome some of the shortcomings of SQLite, which is an older database system. It is an object-oriented and open-source database, which is about 10 times faster than other databases. It is also a rather consistent database system, making it a good choice. It is also a fairly reliable database system. Additionally, they encrypt their data well, helping to secure data. Typically, developers use this for when they need their apps to have high data throughput. This means a lot of data being transferred to and from the database very quickly. Since larger, more involved apps have more data, this is what they typically use. However, Realm charges a fee of $30 per month for hosting data on their servers.

3.9.1.3 AWS

AWS is a set of several different online services provided by Amazon. One group of the services is for their database systems. AWS actually has several database systems that you can access, depending on your needs. For the sake of simplicity, we will group them into one set and evaluate that. AWS database systems are fairly reliable and consistent for retrieving data. They also provide access to use their database services for free, within 20GB of service use. They also encrypt their data well, which makes user data secure. Some of the database systems provided also have quick access times, which is another good feature to consider. One of the drawbacks of AWS is that it is a set of over 200 globally accessible services. Since there are so many services, which all run on Amazon servers, it is typical to see the servers crash and become temporarily unresponsive. This is not good for developing or for the user, since it means that there is a chance that the database service may not work.

3.9.2 Database Comparison

Below is a table to help compare each of the metrics discussed in the previous section. This way, comparing each of them should be much easier.

Table 10. Database Comparison

	Cost	Storage At Given Price	Consistency	Reliability
Firebase	Free	1GB	Consistent	Very Reliable
Realm	$30 / Month	2.5GB	Consistent	Reliable
AWS	Free	20GB	Consistent	Not Always Reliable

3.9.2.1 Cost

Cost is one of the most important factors to consider when deciding on a database to use. Specifically, for this project, we want to pay as little as possible in order to develop our app. We already have to purchase parts for prototyping, as well as parts and PCBs. Naturally, we want to keep the price of each part as low as possible. In this case that would be paying nothing by using the free access versions of each of the database systems. Firebase and AWS are the best options for this, since they offer free access versions. However, Realm requires you to pay, which makes it a poor decision with respect to this metric.

3.9.2.2 Storage At Given Price

Storage at the given price in the table is also very important to consider. This determines how much data we can store on the database while we are using it for this project. The amount we need to use will be very little, since it will mostly involve a user's login information and several preferences that can be represented without using that much data. Regarding the number of users, we will likely only have a small handful of them. This is because we only need the multi-user system to act as a proof of concept. We will not be hosting hundreds and thousands of users on this app, so having a smaller amount of usable data is perfectly fine. With that being said, any of the above options would work well for this since each of them has plenty of data to support our needs.

3.9.2.3 Consistency

This is an important metric to discuss since it involves how accurately we can send and receive data from the database we want to use. This means that, when we write data and read data, we are getting the data we stored in the system properly. Some systems are not very reliable, which makes using them much more difficult. When reliable, the systems are much easier to use and provide exactly what you need, accurate storage. With respect to the options we have explained, all of them are fairly consistent and provide accurate data reading and writing. This means that we will be able to easily create users, log in as a user, and parse data from that user. This will allow us to make the app work properly for any user, which is why it is so important to discuss.

3.9.2.4 Reliability

Reliability is even more important than consistency because it refers to how available the system is to use. When the system is not reliable, that means that it is prone to crashing and not being available. More reliable systems are less prone to this, and are available whenever the user needs them. Regarding the options above, we can see that Firebase is the most reliable, Realm is generally reliable, and AWS is not always reliable. Naturally, this makes Firebase the best option with regard to reliability.

3.9.3 Database Decision

After comparing all of the above options, our team has decided on Firebase as our database system. One of the more important factors that contributed to this choice was reliability. Of the options we explained, it is the most reliable of them and is very available for users to access whenever necessary. Additionally, it offers a free access version, which will keep the overall cost of the project low. Other weighing factors were that it has a large amount of documentation and several useful features. Since none of us have ever used Firebase, the availability of documentation will help us learn how to use the system quickly. The useful features, specifically the ones catered to dealing with users, will be very helpful in allowing us to make the app work in a streamlined way. They will also be helpful for making development easier, since we will not have to implement the features in our app ourselves.

3.10 PCB Comparison

Another one of the main components of our project was which PCB manufacturer we should use to make our PCB. For any project, the PCB is the most important component since it acts as a means to join all of the parts electrically and give them a solid structure to live on. With that in mind, we wanted to choose a PCB manufacturer that works the best for us based on the quality of the PCBs they manufacture. The company we chose was from the company JLCPCB, due to its cost, PCB quality, and shipment availability.

First, we can talk about what the use of the PCB will be in our outlet. Its main use is to hold all of the components together in a secure way. Specifically, all of the components are fixed to a common object instead of being loosely connected by a large number of wires. This is great for making the wired connections more secure as well. It also allows for the components to be more organized, which can be very useful with more complex circuits.

Another one of the main reasons we chose to make a PCB is because it allows for all of the components to be compactly organized in an efficient way. With that being said, it allows us to make the circuitry smaller. This is crucial for our project, since the size of the device is our biggest constraint. For our project to work properly, all of its parts must fit into it in a proper way. Without a PCB, many of the essential parts would not be able to fit into the outlet, which would result in our group having to make the outlet bigger. This is not desirable, since we want it to be as small as possible.

Additionally, the PCB serves to connect each of the components to one another by implementing a circuit. It does this using copper traces and vias to make the wires very thin and flat, reducing a large amount of space. They are also more repliable, since parts are directly connected with solder. This is different from components that are wired together with loose wires usually, since they can use jumper wires with unreliable connections. To that point, if we were to not use a PCB, it makes sense that the connections would also be less reliable.

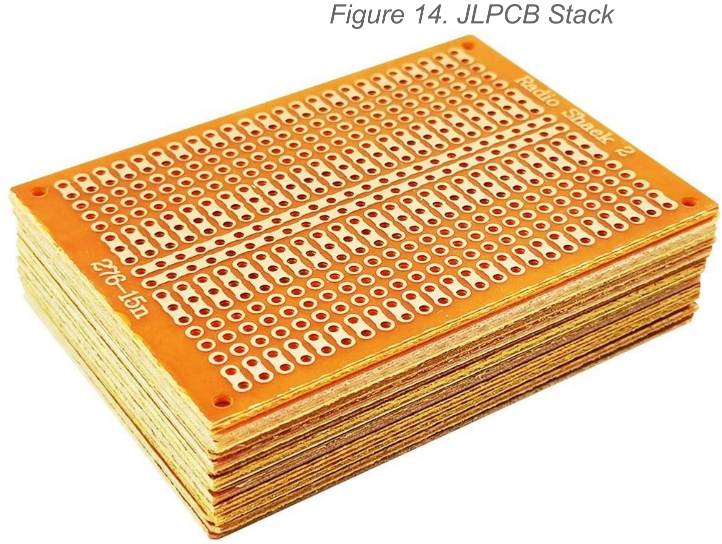
In order to power our PCB and ensure that it will have enough energy for all of its components, the AC to DC converter will perform the task of supplying said voltage to the PCB. By converting alternating current into direct current, the supplied 5 volts will be more than enough for the PCB to perform all the necessary tasks, without the necessity of an extra battery for the device. We are hoping that, by removing the additional cost of a battery having to be replaced by the user, the product will be more desirable in the eyes of a user. This will also make it simpler to use, since the user will not have to put in the work to replace the dead battery.

The PCB company we chose, JLCPCB, will have a cost of around $10 for 5 PCBs. When purchased, the estimated arrival time is approximately 2 weeks. This is relatively quick and is perfect for us, since we plan on starting the hardware construction portion of the project around that time frame. The low cost is certainly a very important part of why we wanted to purchase the PCB from this company. We want to keep our project budget as small as possible. By keeping our costs as low as possible the product we manufacture will also reduce in value, making it more cost effective for a user. This is perfect since the entire point of this project is to reduce cost via cost awareness. However, this does not mean that our project is going to be using the cheapest parts available, causing the project to not work properly. This simply means that we are going to be choosing the cheapest part that won't hinder the overall functionality of the product. That way, we can achieve a nice balance between cost and overall effectiveness of our outlet.

One of the main devices that will be introduced in our PCB will be the relay module. The main use of this part will be to enable the outlet to turn on and off whenever the user decides to. Without these parts in our outlet, the PCB would not be able to turn itself off by itself, hence the need for our group to shift pieces around and achieve the desired function of shutting off the Smart Outlet. The relay module itself is sold by Amazon, and will have an approximate cost of $6.99. One of the main reasons we chose this part was because it fit in perfectly with our PCB, and wouldn't have to sacrifice a lot of space in order to fit in such a device.

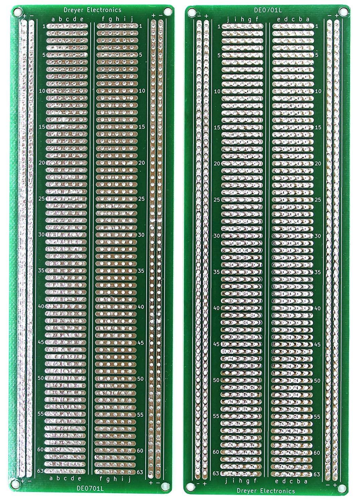
One of the PCBs we had in mind before choosing the right one was the YUN GUI PCB board, which costs around $12 on Amazon. Despite being around the same cost as the desired PCB we had in mind, these boards do not have custom traces and would require you to add additional wires to implement your circuit. Ultimately, due to these reasons, we decided against utilizing this board and sticking to our custom PCB from JLCPCB, which was the perfect fit for our project.

Figure 14. JLPCB Stack



Another one of the PCBs that we had in mind was the DE0701 from Dreyer electronics. Again, Despite being around the same cost as the desired PCB we had in mind, these boards are a little bit shorter than the desired board we had in mind, which could result in some of the pieces not being able to fit into our device. Ultimately, due to these reasons, we decided against utilizing this board and sticking to our custom PCB from JLCPCB, which was the perfect fit for our project.

Figure 15. DE0701 PCB



After all that has been discussed in this topic, it should be clear why we chose to go with the custom PCB from JLCPCB. Despite being around the same price or in some cases a little more expensive than some of its competition, the customization available was a big factor that influenced our decision. One of the most important aspects of this PCB is its high quality material, with high quality traces. These aspects can go a long way in our project, and choosing the right devices is essential for a good functionality of our project.

3.11 Summary after project completion

During our research, we chose to go forward with all the options we listed on our document: the ESP32, the push button, the phone application, the electromechanical relay, the ADS115, etc. During the project's building, the options proved to be the right ones, since our outlet had no major setbacks due to parts conflicts

4.0 Design Constraints and Standards

Every project design has constraints and standards. If one does not uphold standards common for certain types of products, catastrophe will surely unfold. It is important not to cut corners with standards as well as not super proofing everything if it does not need it because that is how costs go up quickly that may not be necessary. Constraints are important as well because those are what define the limitations of our project. It is important to keep these in mind as we continue researching and building.

4.1 Constraints

We are in the beginning of project planning so we will figure out more constraints as we go. Every project needs realistic constraints so that the final outcome is a good one. Each constraint is found out on a case by case basis. Each one has easy ways to be overcome but must be done with thoughtfulness and logic. The main constraint we see right now is the size. Another constraint is sturdiness. Additionally, cost is a major constraint as well. Finally, we see user satisfaction and time as equally important constraints.

4.1.1 Size

One of the biggest roadblocks of our project is size. The bigger size, the more money it will cost to make and more power it might require. Also, if our plug is too big, it might not fit in the wall correctly. For us to meet all the criteria we set out for ourselves, we must be sure to have all the components we have listed. If even one component gets the boot due to size, the device will lose out on some of its functionality. We must be sure to make full use of all our space with our 3D cad design. That will tell us all we need to know. If for whatever reason size does become a real issue that can not be overcome, we will either lose some functionality which we do not consider an option so then we would pivot and instead make our smart outlet into a plug that connects into the already existing outlet instead.

4.1.2 Sturdiness

Sturdiness is a key factor in any design for most products. No one wants a phone that will easily break from a 3-foot fall. The same concept applies here, a smart outlet that is durable and will not break from someone plugging something in too hard or melt from the slight heat of electricity is crucial. That would be very unsatisfactory for the person using the product because then they would have to replace it as well as potentially becoming a hazard.

4.1.3 Cost

The more accurate we want our device to be, the more we will need to spend on our parts for the device. The stronger we want our device to be, the more money we will need to spend. The point is, the better we want our product to be overall, the more money we will need to spend to ensure that outcome. However, each person in the group agreed upon a certain budget price. With that in mind, we must also keep the costs of parts under our agreed budget price. That also keeps our costs low. We do not think that someone would want to spend more than $50 on an outlet. Fifty dollars is already pretty high for an outlet, but we are hoping that people would realize that it is due to it being a smart outlet. Thus we need to keep costs low for that reason as well.

Another point is that, we do not have to have it use the most expensive parts possible. As long as it does what we want it to do, that should be good enough. The simple fact is that we do not need to build a nuke-resistant device that is rated up to 1 billion amps and 1 billion volts is enough explanation. The device just needs to withstand the typical amperage and voltage that usually comes out of an outlet. A big part of our outlet as well is to save people money. It is important that our outlet draws power to be able to function but as well it needs to not use too much power where the outlet ends up costing the user more money. In the long run, this outlet is designed to help the user see where they are spending too much money on electricity and be able to make the correct changes necessary to save money.

4.1.4 Satisfaction of the User

In order to achieve satisfaction of the user, we must make sure to implement the app in a way that makes it easy to use but also provides plenty of information. The connection of the device to the app must also be flawless. This requires bluetooth or wifi connection. We are looking towards the latter to make the app have the best outcome. As long as there is internet in the house, the app will be able to upload the data to the app. Having it use solely bluetooth, means that the person who’s phone it is connecting to would have to be home and also near the outlet. This would limit one of our key features of our outlet being that the user would be able to turn the outlet on or off while not home. As long as the user’s phone can connect to the internet to use the app, that should be no problem.

4.1.5 Time

The time schedule from the beginning of senior design 1 to the end of senior design 2 which is the beginning of August. This gives us a little over 7 months because senior design 1 starts in January. Since our class overlaps with the summer session, we have a little less time than normal since most of the time, students take senior design 1 and 2 during the Fall and Spring semesters. During these 7 months we must design, test, and execute our device to its fullest potential while still hitting all our requirements. We must follow the goals and the timelines set by the professors of senior design as well as the timeline we set for ourselves otherwise we might fall behind. Falling behind is not fun because then we have to play catch-up and we may potentially not finish. That is why it is important to stay on top of our project timeline.

4.2 Standards

Following standards is always important when creating any device but especially when working with electricity that can shock you and can cause a life-threatening injury. Electrical codes are in place to protect the homeowner and the people that live in the home. The codes talk about the wiring as well as how they should be spread out and types of outlets to use where. This section will talk about all the standards associated with wall outlets.

4.2.1 Room Standards

To start with, the standards of wall outlets vary depending on the location of the outlet. The types of outlets change between inside and outside as well. For bathrooms, they have very particular requirements due to the fact that there is water present as well as powerful appliances that are often used in bathrooms like hairdryers.

4.2.1.1 Bathroom Standards

Bathrooms must be able to run a vent fan, light, and a hairdryer all at the same time so sometimes more than one circuit is needed. Thus, the outlet receptacles must be supplied with a 20-amp circuit. That same circuit can be used for the whole bathroom as long as it serves only that room and the vent fans do not contain heaters. If the bathroom does have a vent fan with a heater, the vent fan with heater must have its own 20-amp circuit. Also, to provide light for the bathroom, the lighting must have its own 15- or 20-amp circuit. The receptacle for the outlet needs its own 20-amp circuit. All receptacles in bathrooms must have GFCI protection. GFCI, or ground-fault circuit-interrupter, will be explained more later but basically it is to protect the user from ground-fault electrical shock which can happen when plugging in powerful appliances such as the blow dryer. Finally, the bathroom must have at least one 120-volt outlet within 3 feet of the edge of the bathroom sink.

4.2.1.2 Kitchen Standards

Next is the kitchen which also has very specific requirements as well. Studies show that the kitchen uses the most electricity of any room in the whole house. Kitchens nowadays with updated appliances require at least 7 circuits or more to function. There must be at least two 20-amp 120-volt circuits for outlets in the countertop area for small appliances such as a blender or a slow-cooker. The stove/oven combo needs its own 120/240-volt circuit. The dishwasher and garbage disposal (if the kitchen has one or both) need their own 120-volt circuit and this circuit can be 15-amp or 20-amp depending on the load of each. This can vary depending on the brand or model so that would be according to the manufacturer's recommendations.

The dishwasher circuit requires GFCI protection in case it gets wet and could possibly shock the user. The garbage disposal does not require GFCI protection unless the manufacturer of the product dictates it necessary. The refrigerator and microwave require their own 120-volt circuit individually with an electrical load usually of 20-amp but again it can vary depending on the brand and model and would be according to the manufacturer's recommendations. Almost at the last requirement, we touch on the countertop outlets that are within 6 feet of the sink. These outlets must be GFCI protected as these can also get wet by accident while doing dishes and such. The outlets should be within 4 feet of each other. Finally, the kitchen lighting needs its own 15-amp circuit minimum.

4.2.1.3 Living Room, Dining Room, and Bedroom Standards

The living room, dining room, and bedrooms are next on the list. These rooms are not as strict as the kitchen and bathroom which have probably the two most minimum requirements. These rooms are usually supplied by 120-volt with 15- or 20- amps circuits. These circuits are usually able to provide for more than one room. A wall switch is required at its entryway so that people can turn the light on in the room and see where they are going upon access to the room. The wall switch must be connected to a ceiling light, wall light, or an outlet used for plugging in a lamp that can be switched on and off. The ceiling light must be controllable by the wall switch but it can also have a pull chain. Wall outlets must be no further than 12 feet apart on any wall of the room. Any wall over 2 feet needs a receptacle. Finally, dining rooms must have a separate 20-amp circuit for a single outlet that can be used for appliances like a microwave, entertainment center, or a window air conditioner.

4.2.1.4 Stairway Standards

Next is the stairway. Stairways are not as critical since not all homes, especially in Florida, even have stairs and not a lot of time is spent there. However, it would be very unfortunate to be climbing up or down stairs and suddenly fall all the way down because then someone would hurt themself. For that reason, the only stairway requirement is to have proper lighting. This can be accommodated with 3-way switches at the top and at the bottom of each flight of stairs. This is so that a person can turn the light on or off at the top or the bottom of the stairs. If the staircase is long and has a landing in between the bottom and top, an additional light with a light switch may need to be added at the landing to ensure proper lighting and safety of the individual.

4.2.1.5 Hallway Standards

Hallways are pretty similar to staircases. They are usually long corridors with nothing except walls and doors on either side. Hallways usually also connect most of the rooms in the house with each other and would thus serve as an escape route in the case of an emergency. Especially in a case like this, hallways need to be adequately lighted so that no one trips or hurts themself upon exit or reentry. A hallway over 10-feet-long requires an outlet for general purpose use. Another requirement, similarly to the staircase, is to have 3-way switches at each end of the hallway to allow the lights to be turned on and off from both ends. If there are multiple bedrooms connected by the hallway, multiple switches can be added near the door of each room for ease. That is it for hallways, they definitely have the easiest requirements so far and that is partly because there is not too much danger in a hallway.

4.2.1.6 Closet Standards

Time for the tiniest area in the whole house unless you are a rich millionaire with basically another room for a closet. Closets have quite a few rules and definitely more than hallways surprisingly. One would not think much goes into it but they would be wrong. Most of the rules are in regards to light bulbs. Light fixtures with LED bulbs must be at least 12 inches from storage areas or if the light fixture is recessed from within the ceiling of the closet, then 6 inches is fine. If the light fixture uses incandescent light bulbs that can reach very high temperatures and emit a lot of heat, it must be enclosed with a globe or another type of cover and then it must also be 12 inches from storage areas or 6 inches if recessed just like the LED bulb. Compact fluorescent light bulbs are the easiest and can be within 6 inches of the storage areas recessed or not. All surface-mounted light fixtures have to be located on the ceiling or on the wall above the door if there is space there. This does not include recessed light fixtures.

4.2.1.7 Laundry Room Standards

The arguably 2nd most important room of the whole house is the laundry room. The laundry room requirements vary depending on the washer and dryer but have pretty standard rules regardless. Laundry rooms require no less than one 20-amp circuit for receptacles that would be used for the washer and a gas dryer. An electric dryer demands its own 30-amp with 240-volt circuit and four conductors being part of that. Some outdated homes only have 3 conductors which can suffice for now but new houses being built require 4. Finally, all receptacles need to be GFCI-protected. This is due to the fact that laundry rooms can draw a lot of power when running and sometimes if a washer is broken, water can leak. Both these cases can cause a short. A GFCI outlet would protect the user in both cases.

4.2.1.8 Garage Standards

The last possible room in the whole house is the garage. Some people may not consider it a room but it is, just usually the hottest because they typically do not have air conditioning. That is because it does not have great climate control and nor does it need to. It is a good way to save on cash from not having to cool another room in the house. As of 2017, newly created garages must have at the minimum one 20-amp with 120-volt circuit that is only for the garage. That circuit can be used for any of the needs of the garage such as the wall outlets inside or just outside the garage. There must be at least one switch for using the light inside the garage but for ease someone should put 3-way switches anyways in the case that the person entered the garage on one side and then exits on the other. Garages must contain at least one outlet and add one for each additional car space in the garage. If there are spots for two cars, the garage must have 2 outlets total and so on and so forth. The final requirement of the final room in the house is to have all garage receptacles be GFCI protected. This includes the inside and outside receptacles. Since a lot can be going on in the garage this standard is used to protect the user. Spraying a hose to wash the car and accidentally spraying the outlet, bam a short. Rain from the outside somehow drips into the outlet, bam another short. Maybe drawing too much power using a generator in the garage, bam another short. The user would be protected using a GFCI outlet in all these scenarios.

4.2.1.9 General Room Standards

There are additional requirements in regards to the general electrical codes of the house and are not necessarily pinned on a certain room of the house. All the branch circuits that are for the lighting and receptacles of the house must have AFCI protection. AFCI protection is another method to protect the user from hurt and destruction and is still required in addition to the previous requirements that call for GFCI protection. GFCI protection guards the user but in different ways and are still needed as mentioned. AFCI will be defined more later for better understanding. AFCI protection is enforced as of 2017 and pertains mostly to new construction. Thus, old systems do not require a whole remodel unless the homeowner updates or replaces failing receptacles or devices; then they would be required to add AFCI protection at each spot that is changed.

A couple different ways they can achieve that outcome is first by replacing a standard circuit breaker with an AFCI circuit breaker. Replacing that would make the entire circuit contain AFCI protection. The replacement is usually done by a licensed electrician because this job can get quite dangerous if done improperly. Second, a broken receptacle can be replaced with a AFCI receptacle but that would only protect at that specific location and nowhere else. Where there are rooms that require GFCI protection, there are receptacles that provide protection for both GFCI and AFCI so that would be the replacement item used in places like the bathroom or kitchen. The final general protection is to have tamper-resistant receptacles. The last thing anyone wants is for a small child to be messing around and shocking themself. There are receptacles in place to protect something like that happening called tamper resistant receptacles. This prevents a child from putting an item into the receptacle opening and feeling a nice little surprise.

4.2.2 GFCI Protection

GFCI stands for ground fault circuit interrupters. Annually, hundreds of Americans die due to electric shock. GFCI protection was created to protect individuals from being electrocuted. Some people might think that electrocution only happens during a construction accident or some type of utility accident but these happen in the home more often than expected. Before the requirement of GFCI outlets, about 800 deaths occurred in the United States every year. Now, thanks to GFCI outlets, the number of deaths annually due to electrocution has dropped to close to 200.

Some people, however, confuse the GFCI with a house fuse. The difference is that a house fuse is used to protect a structure from an electrical fire. When a hot wire touches a neutral wire for whatever reason, a large amount of current will flow through the circuit and heat it up immensely. This is when the fire can occur, thus the house fuse heats up quicker than the wire and burns out before a fire can start. GFCI is different from the house fuse in that it is implemented into the outlet itself. When someone plugs something into the GFCI outlet, it tracks how much power is going into the device such as a blow dryer in the bathroom. Let's also say that the sink is full of water and now the person drops the blow dryer in the sink. In normal circumstances the person would now be electrocuted and be very seriously injured. Due to the GFCI outlet though, it would feel the interruption in the current and stop the power. This can prevent serious injury or even death. All the electric current cares to do at that moment is flow through the body to get to the ground which means current would be flowing through the whole body if not for the GFCI outlet.

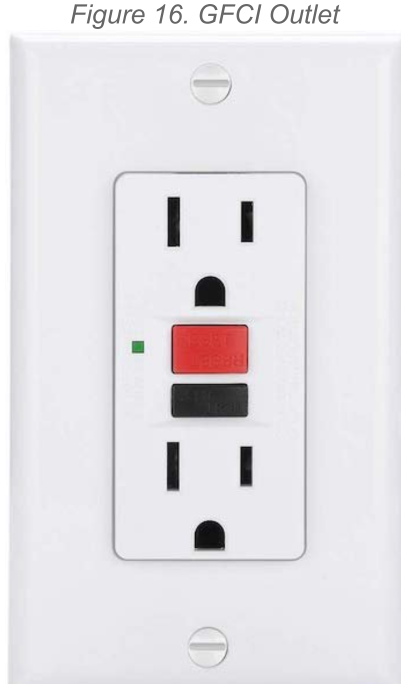
4.2.2.1 GFCI Outlet vs Normal Outlet

Some other key differences of a GFCI outlet versus a normal outlet are that GFCI outlets have a test and reset button on the outlet itself. Pushing the reset button should enable power to the outlet and offer the GFCI protection. If the test button is pressed, power to the outlet is cut off. In situations where someone may be shocked, the outlet hits the button itself and cuts the power off. In a normal 120-volt outlet in the United States, there are two slits and a hole centrally beneath them. The left hole is slightly larger than the one on the right. The left slit is the neutral hole and the right slit is the hot hole. The hole beneath the two is the ground hole. When appliances are functioning properly, the flow of electricity is from the left hot hole to the right neutral hole. A GFCI outlet will record the flow of current from the hot hole to the neutral hole. If there is an imbalance for even a millisecond, the GFCI outlet will trip the circuit. The imbalance can be as small as 4 to 5 milliamps. It needs this quick reaction time as well as the small change because at 10 milliamps, the muscles of the human body begin to freeze from electric shock which means that the person can not even let go of the thing that is hurting them. This is even more crucial because two seconds in a state like that can cause death.

4.2.2.2 Further Clarification GFCI Outlet

The way the GFCI outlet detects the change in current and cuts the power was previously mentioned but we will explain it just a bit further. It detects the change of current from the hot hole to the neutral hole. Where does the difference go? The difference would be going through the person which is no good. That's how it knows to cut the power off. Finally, GFCI outlets do not last forever and usually last about 10 years but that can vary. Some experts suggest testing the GFCI outlet once per month to make sure it is working properly. Below is a GFCI protection outlet. It has the usual test and reset buttons previously talked about.

Figure 16. GFCI Outlet



4.2.3 AFCI Protection

AFCI stands for arc fault circuit interrupters. To explain what AFCI protection is, we must first explain what an arc is. Arc refers to when loose or corroded wires at the connection points sporadically touch which causes an electrical spark between the contact points of the metal wires. This can also be referred to as an arc. When someone turns on a light switch or an outlet and a faint buzzing noise or hiss can be heard, that is the sign of an arc occurring. When an arc occurs, this means heat is produced which can melt the insulation around the wires which could then cause an electrical fire. However, just because a noise can be heard, that does not mean a fire will occur. That still means there is a danger and the noise should be assessed.

4.2.3.1 Different Types of Faults

An arc fault, ground fault, and short circuit are all different things but sometimes get confused for the same thing because they all have similar outcomes.

4.2.3.1.1 Short Circuit

A short circuit, which almost all engineers should understand, is any situation where the current from the “hot” hole strays from the wiring circuit and makes connection either with the neutral wiring or the grounding pathway. When this occurs, the flow of current has no resistance and the current increases dramatically. When something like this happens, there is too much current for the circuit breaker to handle and normally it will trip to try to stop the flow of current before total death and destruction occurs.

4.2.3.1.2 Ground Fault

A ground fault is a type of short circuit but is specifically when the hot wire makes connection with the ground. A ground fault is sometimes called a “short-to-ground.” Just like the short circuit, loss of resistance and extremely high current causes the circuit breaker to trip. In this case, the circuit breaker may be too slow to react and damage may have already been done before the circuit breaker can react. Thus, this is where GFCI outlets are used as previously mentioned in places like bathrooms, kitchens, or outside where water may be prevalent.

4.2.3.1.3 Arc Fault

An arc fault is when wired connections or wires that are corroded cause sparking which is also called arcing. This phenomenon can also occur when a tree falls on a power line because of a storm and the line is now on the ground. The system lets out a lot of current between the conductors and or to the ground. Basically, an arc fault in simplest terms would just be when a current takes a path it was not supposed to and is unplanned.

When the current takes the unintended path, high heat is produced and this can lead to a fire. This type of fire is usually called an electrical fire. The temperature of a fire such as this can reach up to or even over 10,000 degrees fahrenheit.

Thus, since this is such a dangerous outcome that can occur, preventatives are needed to stop this from ever happening. A big concern with arc faults is that when a fire is created, it is often out of sight and it could be located in the walls or in the ceiling. This means that the fire would spread throughout the whole house before a fire alarm could even do anything. This can lead to the whole house coming down and causing serious damage to the house and everyone involved. This is why arc fault circuit interrupters are a requirement in today's standards of living. It can and does save many lives.

4.2.3.1.4 AFCI Protection Explained

As previously stated, AFCI protection is not to be confused with GFCI protection. Arc fault protection is defined as any device that is made to protect against bad connections that are producing arcing or sparking. The device detects an electrical arc and cuts the power from the circuit to stop an electrical fire from occurring. However, it is not just as simple as the AFCI protection detecting an arc and flip off power and done. Some electronics such as a motor driven vacuum cleaner or a furnace motor actually create arc as a natural byproduct of them being used. Another normal occurrence of an arc being produced is when a light switch is turned off and the disconnecting of the connection causes an arc for a split second. In scenarios such as these, the AFCI protection must be able to recognize the difference between a real threat and what is just a typical routine. Manufacturers of AFCI protection devices simulate their products under hundreds of different scenarios so that way the device can tell the difference between a potentially life-threatening arc and a normal everyday arc.

4.2.3.1.5 Different types of AFCI Protection

There are a few different types of AFCI protection. For an AFCI protection device to be considered for use, it must be a listed product of the NEC®. That means an AFCI protection device must undergo vigorous testing in labs to meet the important standards for AFCIs, seeing as they save lives. So, the three most common that have passed much testing are the branch/feeder breaker AFCI, combination breaker AFCI, and breaker AFCI and GFCI protection.

4.2.3.1.6 Branch/Feeder Breaker AFCI

The first type of AFCI protection device we will be talking about is the branch/feeder breaker AFCI. The branch/feeder breaker AFCI device is supposed to be installed at the base of the branch circuit or feeder (hence the name) at a location like the panelboard. This device provides protection for detecting arc faults that happen due to line-to-line, line-to-neutral, or line-to-ground connections. Older homes have a problem using this type of AFCI protection due to the fact that they sometimes have outdated electrical foundations due to old codes. In older homes, the circuits oftentimes cannot support shared neutral circuits. Thus in situations like these, a two-pole AFCI is used. This simulates three-wire circuits that are often used in these kinds of implementations.

4.2.3.1.7 Combination Breaker AFCI

The next type of AFCI protection device we will be focusing on is the combination breaker AFCI. Combination breaker AFCI is often used in conjunction with the branch/feeder breaker AFCI. The combination breaker AFCI is useful to detect an arc that is occurring as little as 5amps. It can observe this phenomenon in branch circuits and power supply cords. This is another feature that is not often found in older homes unless remodeling or updates are done to the circuitry. This is required as of the beginning of 2020.

4.2.3.1.8 Breaker AFCI and GFCI Protection

The final type of AFCI protection device we will be discussing is the breaker AFCI and GFCI protection. This is usually achieved by using an AFCI circuit breaker and a GFCI receptacle. That way, the user is protected from an arc fault and a 5mA ground fault. There are also devices that are 2-in-1 and offer GFCI and AFCI protection. This is useful in cases such as the bathroom or kitchen where water can easily come into contact with outlets so GFCI outlets are not only required but necessary and additional protection is sought out. A person can never be too safe so it is always a good idea to have both protections if possible. Below is what an AFCI protection device looks like that is connected to the circuit breaker.

Figure 17. Breaker AFCI

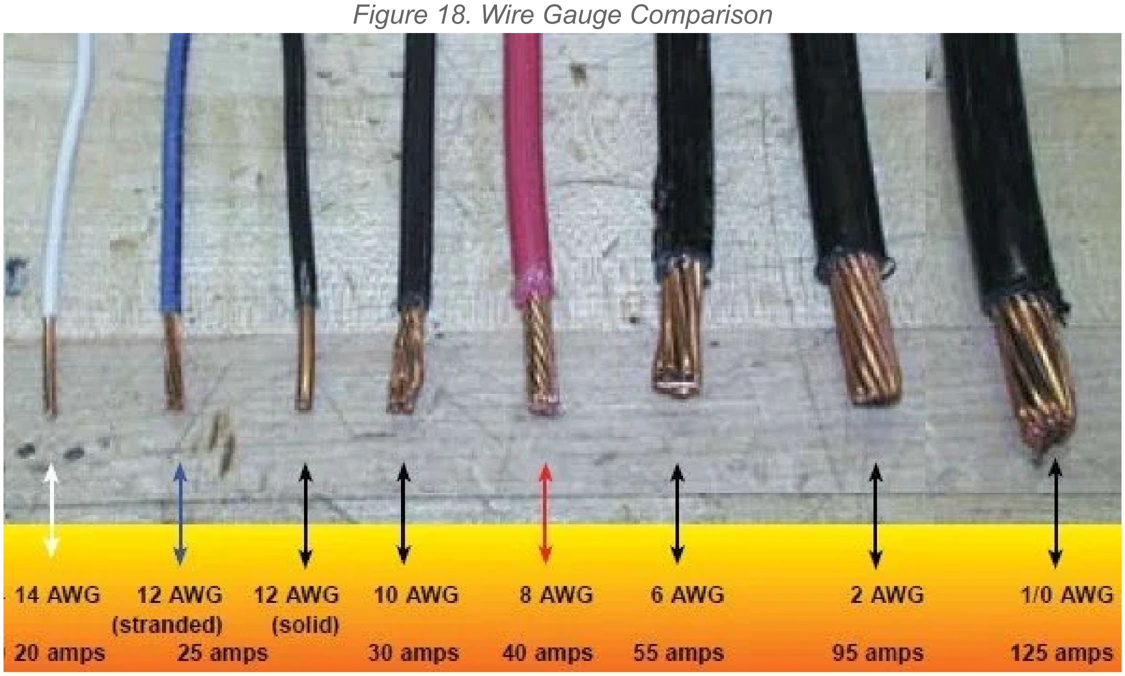
4.2.4 Cable and Wiring

One basic component that people sometimes forget about is the actual cable and wiring that connect all the electrical circuitry of the house. To start with there is actually a difference between what a wire is and what a cable is. Cables are a gathering of two or more wires in a single jacket and a jacket is basically a plastic tube covering. Wires are the actual metallic conductors usually consisting of steel, copper, aluminum, and silver. We will explain the different types of wires and the different types of cables.

4.2.4.1 Wires

Wires come in a few different sizes which is also called gauge for wires. Usually you measure the size going in reverse order. So a 14-gauge wire is actually smaller than a 12-gauge wire. The "higher" the gauge like the 14-gauge, the less current that can flow through the wire. The "lower" the gauge like the 12-gauge, the more current that can flow through it. The normal sizes to find in homes are 12-gauge and 14-gauge wires. Strong and powerful appliances such as the electric stove, electric water heater, electric dryer, and A/C unit will almost always use 10-gauge, 8-gauge, or 6-gauge wires. If someone wishes to add an outlet to a wall, that person would need to make sure to use the same size gauge as what is already in the circuit. Otherwise, the outlet may not function properly if at all. Below is a great depiction of some different size wires by the American Wire Gauge system. It also depicts how much current each size can wield.

Figure 18. Wire Gauge Comparison



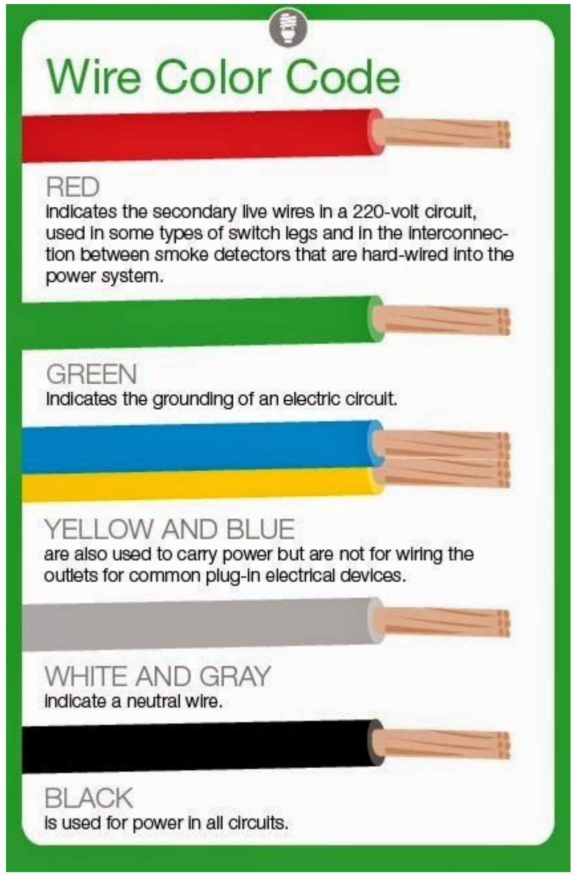
4.2.4.2 Wires - Stranded vs solid

As can be seen in the above figure, the 14-gauge wire is solid whereas the 2-Gauge wire is stranded. There are benefits in different situations in using one over the other. Stranded wire is a bit more flexible due to the nature of multiple strings of metal winded together versus a single strand of copper that will be more brittle and will snap easier. The stranded wire keeps itself together by being binded with multiple strings. The solid wire does not have anything to help keep it together other than itself. Anyways, if someone wishes to pull a wire through a conduit, stranded wire is the right choice because it will need to be flexible to take all the turns a conduit contains. However, if pushing through a conduit, it is best to have a solid wire for that job.

4.2.4.3 Cables

Electrical cables have a few different denominations. They are sorted by two numbers connected by a hyphen. So, a cable that is 14-2 has 2 14-gauge wires inside with one being the hot wire and the other being the neutral wire. Cables can also carry a bare copper wire that serves as the ground. Each conductor is color-coded to help the person realize which conductor does what in the circuit. Cables themselves are even color-coded to make it even easier for the user. A white cable means it is a 14-gauge wire that can do 15-amps. A yellow cable means it is a 12-gauge wire that can do 20-amps. An orange cable means it is a 10-gauge wire that can do 30-amps. A black cable means it is either an 8-gauge or 6-gauge wire that can do 45-amps or 60-amps. If the user checks the plastic coating of the cable, the specifics of if the black cable is 8 or 6 and 45 or 60 is specified there. The last cable color, gray, is an underground cable that is meant to be put beneath the earth's surface and its specifics of gauge and amperage rating is usually written on the plastic coating of the cable. Just because color coding exists does not mean it always did. Color coding cables is voluntary and did not even start until around 2001. Below is the color code chart for the conductors in the cables better known as the wires.

Figure 19. Wire Color Code



4.2.5 The Application

In order to have successful use of our device, the app is of utmost importance. However, there is a lot behind the scenes to make sure the phone app is a success. From adequate internet connection via wifi and proper coding of the application, there are many standards to be followed for the application.

4.2.5.1 Javascript

Javascript is what we are coding our app behind the scenes. Javascript is a dynamic programming language that is usually used for web development, web applications, game development, as well as many other projects as well. Javascript is usually for doing bold things on the web. As one might notice though, our app is not necessarily on the web but on a phone app itself. That is ok because javascript is also great for phone applications as well. Javascript is good for adding click to reveal dropdown menus, adding more information to existing pages, and changing the colors that are on the page. If Javascript did not exist, web pages would simply consist of HTML and CSS.

4.2.5.1.1 HTML

HTML stands for Hyper Text Markup Language. It is the code that is used to structure web pages and their information. Information can be structured in a variety of different ways such as in paragraph form, in a bulleted list, or with the use of images and tables. HTML works by defining the structures of the information on the web page and is basically a markup language, hence its name. The coder uses elements of the code to enclose or wrap parts of the information how they would like for it to be arranged or displayed. Using enclosing elements of the code can make a word image give a link to somewhere else or change the words in some way such as italicize or make the font larger or smaller but this is mostly done in conjunction with CSS. To sum up HTML, it makes the words on screen pretty to the eyes and is important because looks are everything when looking at a webpage.

4.2.5.1.2 CSS

CSS stands for Cascading Stylesheets. Basically, HTML and CSS work hand in hand. To use CSS however, one must first know how to use HTML because it is a building block for CSS. CSS is mostly used to style and create the layout of your information. Cascading Stylesheets is used to change the font, color, size, and spacing as well as building on animations or other decorative features. It can also divide the information up into separate columns or rows. Basically, HTML is for the structure and CSS is for the looks. Below is a picture that further illustrates the differences of each and helps to further explains HTML, CSS, and Javascript.

Figure 20. Code Structure Illustration

4.2.5.1.3 JSX

To be more precise, the kind of javascript language we will be using for our project is JSX. JSX stands for Javascript XML. It is an extension of Javascript that allows the coder to straight up write HTML in the Javascript code. Javascript

XML is faster than normal Javascript because it performs optimizations when converting over to normal Javascript. JSX uses components that compartmentalize markup and logic into separate files. This helps speed up the process tremendously. A few reasons why JSX is good is that it is easier to add HTML in JSX, can easily change HTML components to JSX elements, can prevent some forms of hacking, and it is fairly easy to correct errors as most errors are normally found at the time of compilation. Some reasons JSX is bad is that errors in the HTML throw out an error message and HTML code must be typed in a specific way otherwise an error message will occur. Basically, JSX is much better than normal Javascript, however, errors arise much easier with the HTML on Javascript XML.

4.2.5.1.4 XML

All this talk about this special version of Javascript without an explanation on what XML really is. XML stands for Extensible Markup Language. XML is a language that is very similar to HTML but it does not use the predefined tags that HTML uses. Instead, the coder makes the tags that they need and then uses those instead. It is nice because the person does not need to remember all the tags or look up tags that they need to use. Instead they basically make up their own tags and use them as necessary. XML also has a standard format so if the XML data is shared across platforms via the internet or by other means, the person receiving the data can still make use of it due to the standardization that it has.

4.2.5.1.5 Linter

One final important component of our code is the linter. Linter comes from the original tool that was called Lint. The original Lint was intended for examining C-based codes. Stephen Johnson is the person who created Lint when he still worked at Bell Labs in 1978. Linter helps to improve the overall efficiency of the code. Linters are not only used for Javascript but most people that use them make use of them for Javascript codes. Linters help the code by reading over the code and finding any problems that the code might have.

Linters are not exclusive to only compiled languages as it also works for interpreted languages and it could be seen as more useful in those situations because interpreted languages most of the time do not have a compiler to find the errors. A few pros of using a Linter are fewer errors in the code, can ensure better safety from hacks, and can help other coders learn about the code quality and to help them make changes or additions. There are not really any cons to having a Linter other than having to add it to your code. Linters' most common problems that they detect are syntax errors. Linters can be downloaded from the internet and implemented into the code of certain problems that the coder needs fixed. If the coder can not find one of those problems on the internet, the coder can simply make a linter themself for the specific problem that they need fixed and problem solved.

4.2.5.1.6 Github

The final piece of the puzzle is GitHub. GitHub is where we will be storing code while developing our app for use on android and apple phones. GitHub is a code hosting platform for version control so that people can work on coding projects together. It is very handy because it is similar to a google document where if everyone has permission to view it, everyone can work on it together in real time and people can see when changes are made and are updated almost instantly. GitHub is great because it is free and open source.

4.3 Summary after project completion

After concluding our project, we can confirm that some of our fears came to be true. One of our major constraints was size, given the fact that our project had many essential parts, we had to build a 3D printed case in order to fit all of them in, giving us an extra roadblock during the design of it. When it comes to standards, we always kept them in mind, installing safety nets for the user’s protection if something goes awry.

5.0 Project Design

This section will contain specific details in how the design of the device will work in regards to both software and hardware.

5.1 Hardware Design

The design of the hardware is a key component within our project. Although this is not final it will allow a clearer picture of what we expect regarding physical design as well as layout of some of our materials.

For our project, the Smart Outlet will be designed as an extension to an outlet, meaning that the user will have to plug the device into the desired outlet. After the installation is complete, the device will start accepting voltage, so that it can power itself up. Without the voltage, the device will not work at all, since we will not be installing its own power source due to size constraints. If the device is on, a light will be turned on in the device, indicating to the user that they have installed the Smart Outlet correctly and that it is ready to start measuring voltage and current. It is important to affirm that every single button and light will have a tag above or below it, indicating to the user exactly what the function of said switch is.

The device will have two buttons in its interface, and they will have two different functions: the first one will serve the function of powering on or off the device, depending on the state of the Smart Outlet when pressed. The second one will start pairing the Smart Outlet with the user's smartphone device. They will be clearly tagged with its function, to make sure the user doesn't confuse them when utilizing the device.

First button in the device would be the on button. When the user clicks on it, the device will power on, and start measuring current and voltage. For that button to work, however, the device must be connected to a power source. If the user presses that button when not connected to an outlet, nothing will happen.

Another use of that button would be the off option. When the user clicks on it and the device is turned on, it will automatically shut off power and stop the supply of voltage and current.hen the user clicks on it, the device will power on, and start measuring current and voltage. For this button to work, however, the device must be connected to a power source. If the user presses that button when not connected to an outlet, nothing will happen.

The second button will be used to start pairing the device with the application. By clicking on it, the device will start emitting a flashing blue light, indicating that it is ready to be paired with a smartphone. When the user finally connects their smartphone device to the Smart Outlet, the light will stop flashing and instead emit a steady blue light, indicating to the user that the pairing was successful. For this button to work, however, the device must be connected to a power source. If the user presses that button when not connected to an outlet, nothing will happen. Also, if the device is already connected to a smartphone device, the button will not start pairing up again. For that to happen, the user will have to unlink his device from the phone in the user's application. After that is done, the light will turn off and the device will be ready to be paired up again.

The device will have three lights in its interface, and they will have three different functions: the first one will serve the function of showing if the device is on or off, depending on the state of the Smart Outlet. It will emit a yellow or a red light, depending on the status of the device. The second one will show if the device is connected to a smartphone device, and will emit a steady blue light or a flashing blue light, depending on the status of the device. The third one will emit a green light, and will indicate when a timer is currently on. They will be clearly tagged with its function, to make sure the user doesn't confuse them when utilizing the device.

The first light in our device will serve the purpose of indicating if the Smart Outlet is on or not. If the light is on and shows a steady yellow light, it means that the device is on and supplying power to a desired electronic device. If the light is on but displaying a red light instead, it means that the device is on, but not supplying power to an electronic device. If the light is off, it means that the device is not on and not supplying power to an electronic device. For this light to work, however, the device must be connected to a power supply. If the Smart Outlet is not plugged to an outlet, the light will stay off.

The second light in our device will serve the purpose of indicating if the Smart Outlet is connected to a device via bluetooth or not. If the light is on and indicating a steady blue light, it means that the device is correctly turned on and connected to a smartphone device. If the light is on and indicating a flashing blue light, it means that device is in the process of pairing up with a phone device. If the light is off, it indicates that either the device is turned off or that the Smart Outlet is currently not paired up with any smartphone device. For this light to work, however, the device must be connected to a power supply. If the Smart Outlet is not plugged to an outlet, the light will stay off.

The last light in our device will serve the purpose of indicating if there is a timer in place for our device. It will emit a steady green light, and it will stay on as long as the timer is still going. If the timer ends or the user cancels the timer, the light will immediately shut off. If the light is off, it indicates that either the device is turned off or that the Smart Outlet doesn't currently have a set timer. For this light to work, however, the device must be connected to a power supply. If the Smart Outlet is not plugged to an outlet, the light will stay off.

5.1.1 Hardware Design Methodology

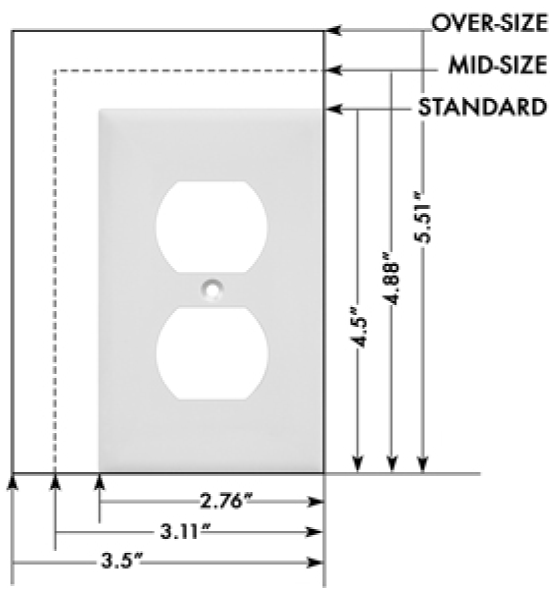
The most important components of our design are what we focus on, and these are the current transformer and the voltage sensor. However these are not the only aspects of the design in which we have to worry about. Our outlet is controlled through the creation of a PCB. The board will have to be able to handle distribution of power, readings of information coming from the sensors, and then transmitting that information through wireless communication methods. Furthermore, the board should be able to receive information from the user regarding turning it off and on. The brain of this operation is going to be the ESP32. It is what will communicate between the different hardware components of our project.

Apart from the components we also must touch on the physical design that our project is to be built to. We must take into account the size of the hardware we are working with as an outlet is a small enclosed place that will need to be able to house all of our components while remaining safe from any possible disturbances that may affect it.

5.1.2 Housing Design

In this section we will be going over the expected housing of the outlet. Furthermore, we will also discuss our expectations for how it will be constructed. As stated before there are many different types of possible outlets when working with consumers. Our design will focus on the more basic one found in the homes of every American, a Nema 1 of type B type of receptacle outlet. More specifically we will use a grounded Nema 1 outlet type. All of type B outlets have slots of different sizes with the taller one being the neutral connection and therefore we will be using that standard as well. Our outlet prototype will not be used in an area of high probability of shock such as a kitchen or bathroom and will therefore not have the GFCI or ground fault circuit interrupter that is required by the electric code. The reason for this is that we have the constraints of being a student with limited time and finances in order to accomplish this national safety standard within this limited semester. An example of what the outlet could look like is shown below.

Figure 21. Housing Design Reference



There are a few different types of outlet wall plate covers which are thereby related to the size in which our outlet needs to be. Furthermore when comparing to the electrical boxes available for consumer purchase we can get a rough estimate of our sizing for the boxes. Below are the references for the most common electrical boxes available.

Table 11. Common Sizes for Electrical Boxes

Box Dimensions (Inches)	**Minimum Cu. In.**	**Max # Conductors AWG 14**	**Max # Conductors AWG 12**
3 x 2 x 1 1/2"	7.5 cu. in.	3	3
3 x 2 x 2 "	10	5	4
3 x 2 x 2 1/2"	12.5	6	5
3 x 2 x 3 1/2"	18	9	8
4 x 2 1/2 x 1 1/2"	10.3	5	4
4 x 2 1/2 x 1 7/8"	13	6	5
4 x 2 1/2 x 2 1/8"	18	7	6
4 x 1 1/4" square	18	9	8
4 x 2 1/4" square	21	10	9
4 x 2 1/4" square	30.3	15	13

From this information we can gather that a good size for our outlet that would fit into most normal wall spaces is around 4 x 2.5 x 2 inches. These dimensions offer a nice middle ground between the total size of the wall space, the size of the outlet, and the clearance we expect to need for our company. The material for our prototype will most likely be made using some form of wood at the commencement of the project as it is affordable and easy to be manipulated to different sizes. The dimensions of our electrical box are dependent on the components we are using and the most important component that we have to consider is the current transformer as it is the largest component that we have within our system. We want the dimensions of our box to be available to be used in most outlet spaces and crammed locations like New York City are known to have a smaller outlet clearance than Orlando. An example of what our enclosure would look like is shown below. When compared to other models currently on the market, ours will appear to be more chunky in order to account for the fact that we are using off the shelf components rather than custom made for us.

Figure 22. Housing Design Example

5.1.3 Part Schematics

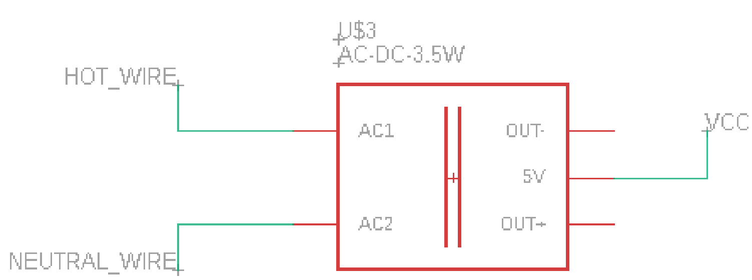
This section is an overview of the different parts and component schematics that will be on the final PCB. This is not currently final but is necessary to get a better understanding of everything and how it will all work together in a small enclosure.

5.1.3.1 Power Components

The AC to DC converter is perhaps the most important part in our system. It is what is going to allow us to power the different components of our pcb. As of right now we are using a module ac to dc converter for safety as we want to make sure we do not mess any of our other components as working with ac has a possibility of being dangerous. If time allows it, we will design our own pcb using the TI workbench and optimize it for use within our system.

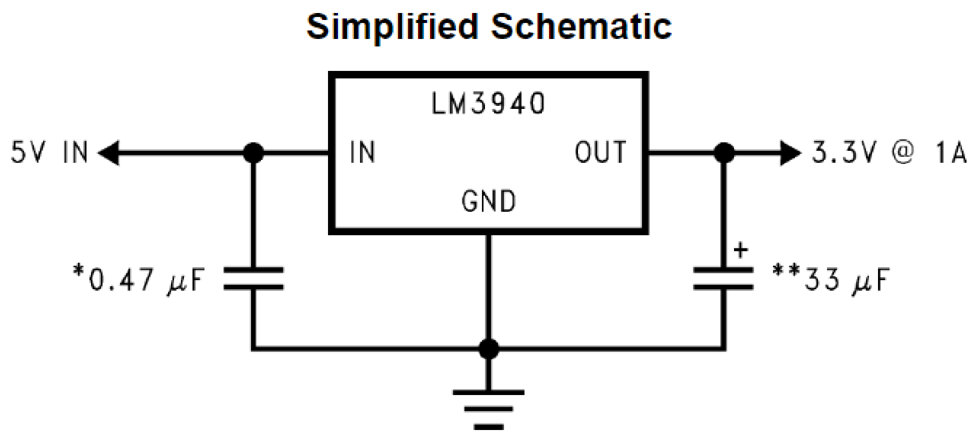
The AC to DC transformer takes the 120 RMS voltage from the wall into the hot wire and steps it down to a constant 5 volts. This 5 volts is then able to be converted into whatever is necessary for the other components.

Figure 23. AC to DC Converter Schematic



The 5 volts will be converted to 3.3 for some components such as the ESP32 as that requires a Vdd of 3.3 in order to function correctly and can be damaged if that is exceeded. In order to get 3.3 we will use a regulator in order to provide a stable source of power for the devices which require it. The reason we will utilize a regulator is that a voltage divider has the possibility of things going bad as it is not supposed to draw current. It is only used during high-impedance connections and with our low voltage, we wouldn't be utilizing high impedances as so. We will be using the TILM3940. An example of this schematic is shown below.

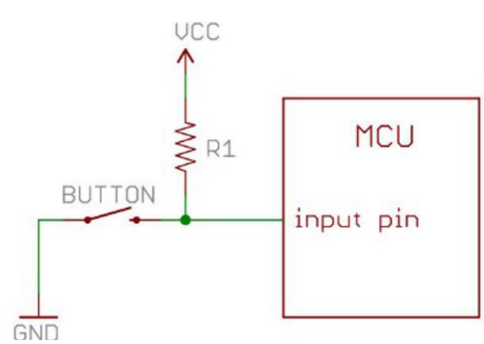
Figure 24. Voltage Regulator



5.1.3.2 Button

Our circuit will have a manual component which can be controlled by the user in order for the relay to turn on and off. The button will be connected to the MCU and will be a simple circuit straight to ground when pressed. The actual circuit will be controlled by the relay and therefore this does not need to care about the high current ratings.

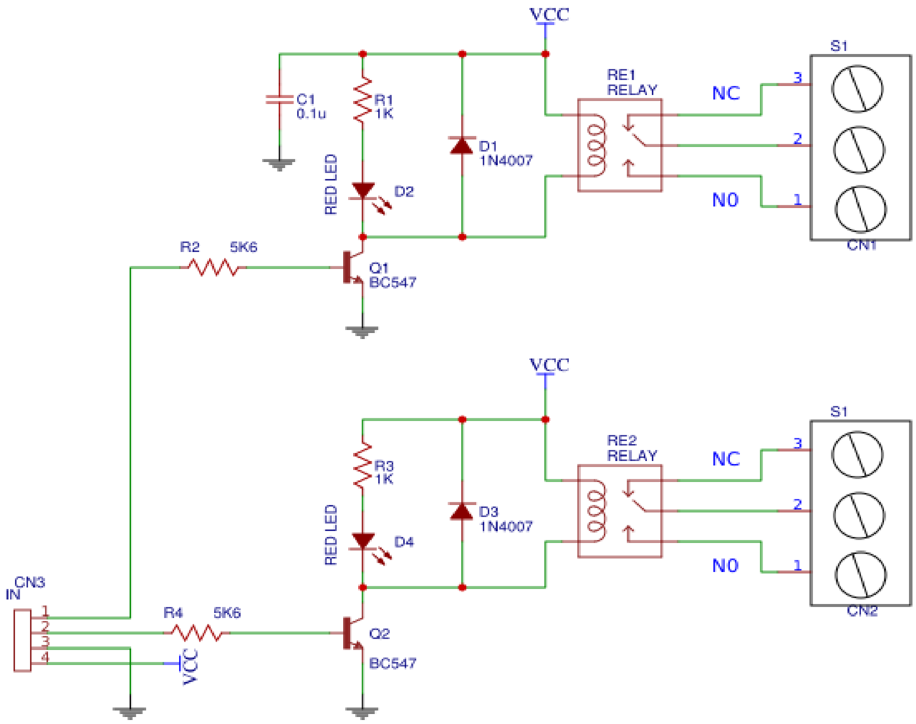
Figure 25. Button Schematic



5.1.3.3 Relay

A 2 channel 5V Relay system will be used in order to control the voltage going to the devices. This relay will be after the ac current has been stepped down and converted. This relay is negative logic and will change when the Vin pins are connected to gnd. This will be done from the MCU. The relay will be able to be controlled by the user in the way that they can turn it on and off from the push off the button after being processed from the MCU or the user will turn off the outlet from the app connection and that connection will be processed from the MCU to turn off the relay. The relay schematic is shown below.

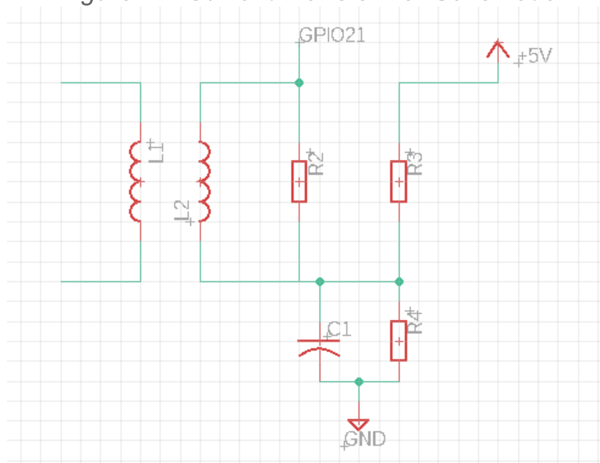
Figure 26. Relay Schematic



5.1.3.4 Current Transformer

The AC current transformer will allow us to utilize it as a current sensor as it is the sealed secondary of a current transformer. The current transformer will give us a real time measuring of the current flowing through the device connected to our outlet and in tangent with our analog to digital. The current transformer has 2000 turns.

Figure 27. Current Transformer Schematic



5.1.3.5 Voltage Transformer

In order to measure the voltage from the ac signal we will have to transform it and step it down. This transformed signal will go into the ADS1115 ADDC and we will be able to read it from there. Below is a simple schematic utilizing the ZMPT101B voltage transformer. We are not completely positive in the validity of this simple circuit and may transition to the more complicated one utilizing LM358 op amps in order to increase safety.

Figure 28.Voltage Transformer ZMPT101B

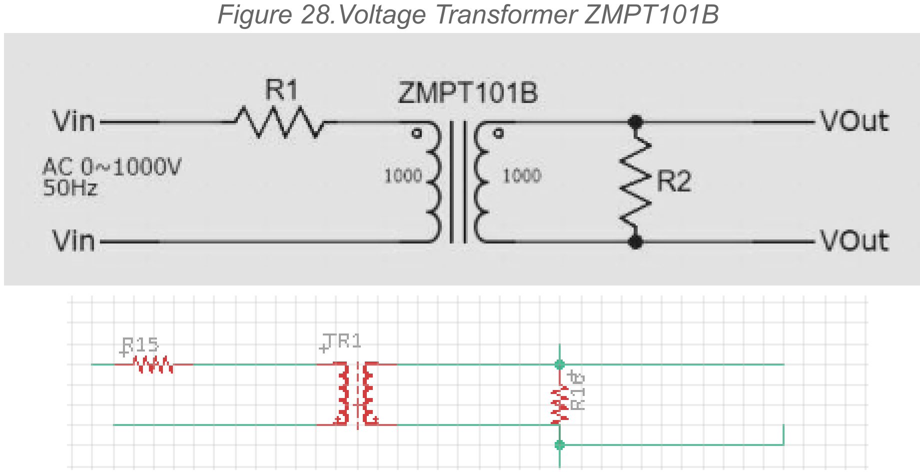
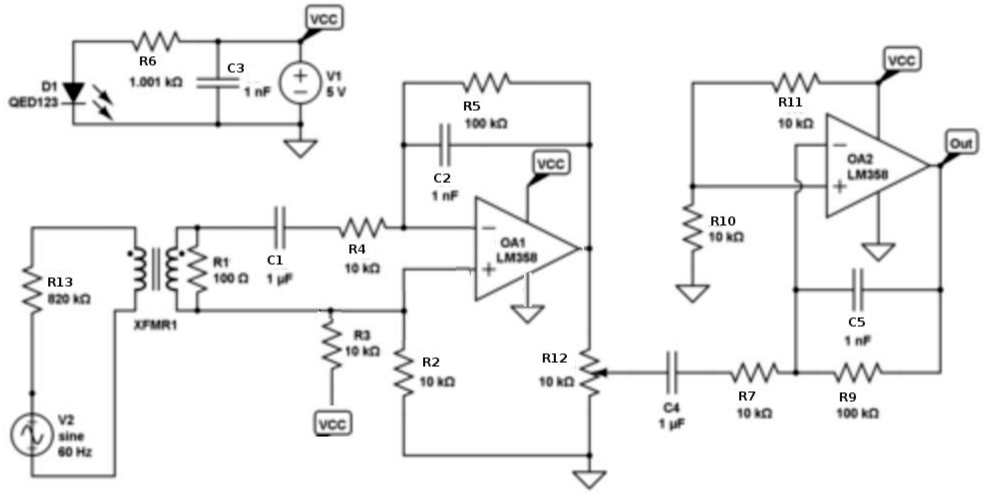


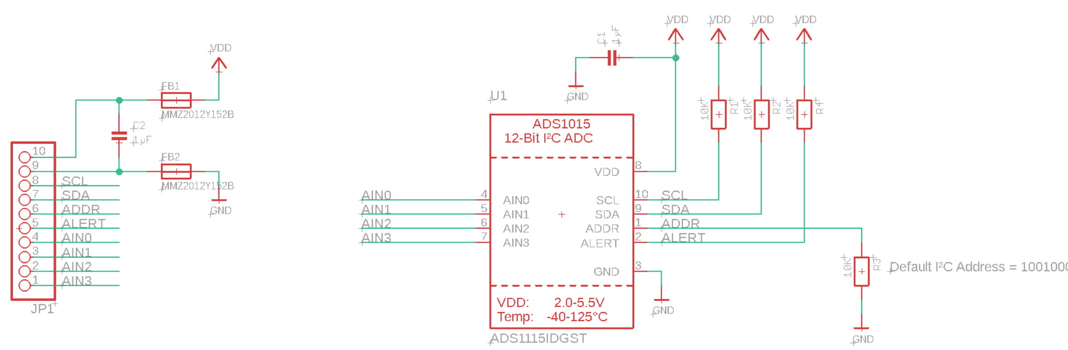
Figure 29. Voltage Transformer ZMPT101b and LM358



5.1.3.6 Analog to Digital Converter

The ADS1115 will be the adc used in order to measure the voltage differential between the voltage being used and the ground. This in tangent with our current measuring components will allow us to calculate the power used within the system. Below is a general schematic of the ADS1115. We will most likely get rid of the pins we do not find necessary in order to keep our board compact.

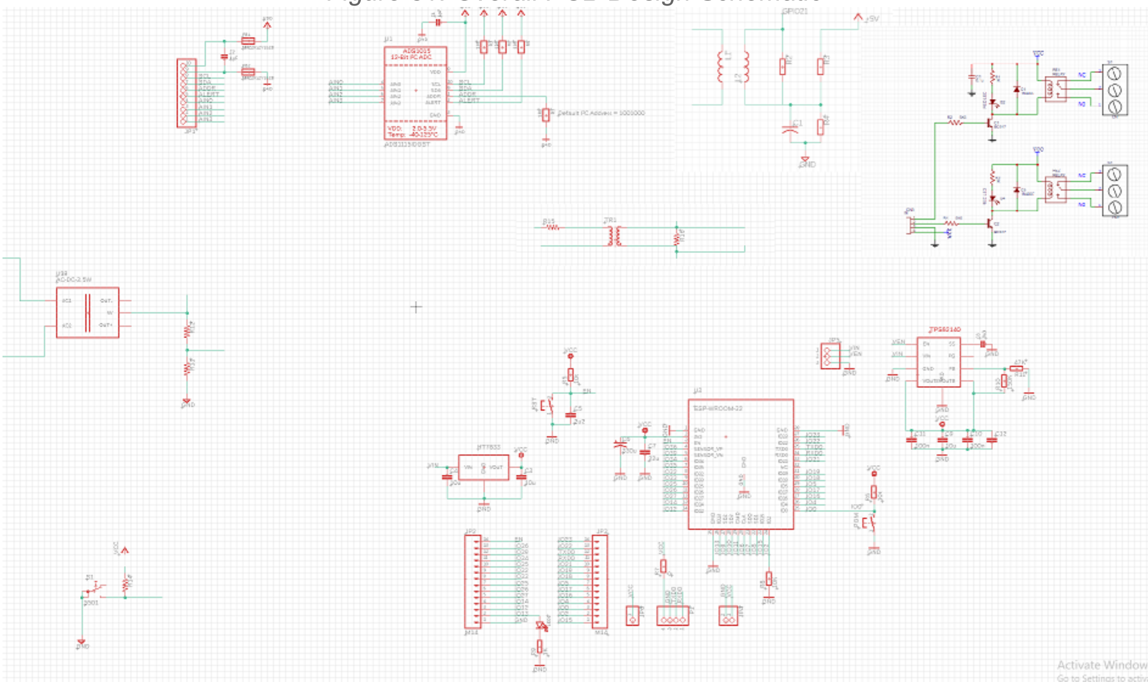
Figure 30. Analog to Digital Converter Schematic



5.1.3.7 Overall PCB Design Schematic

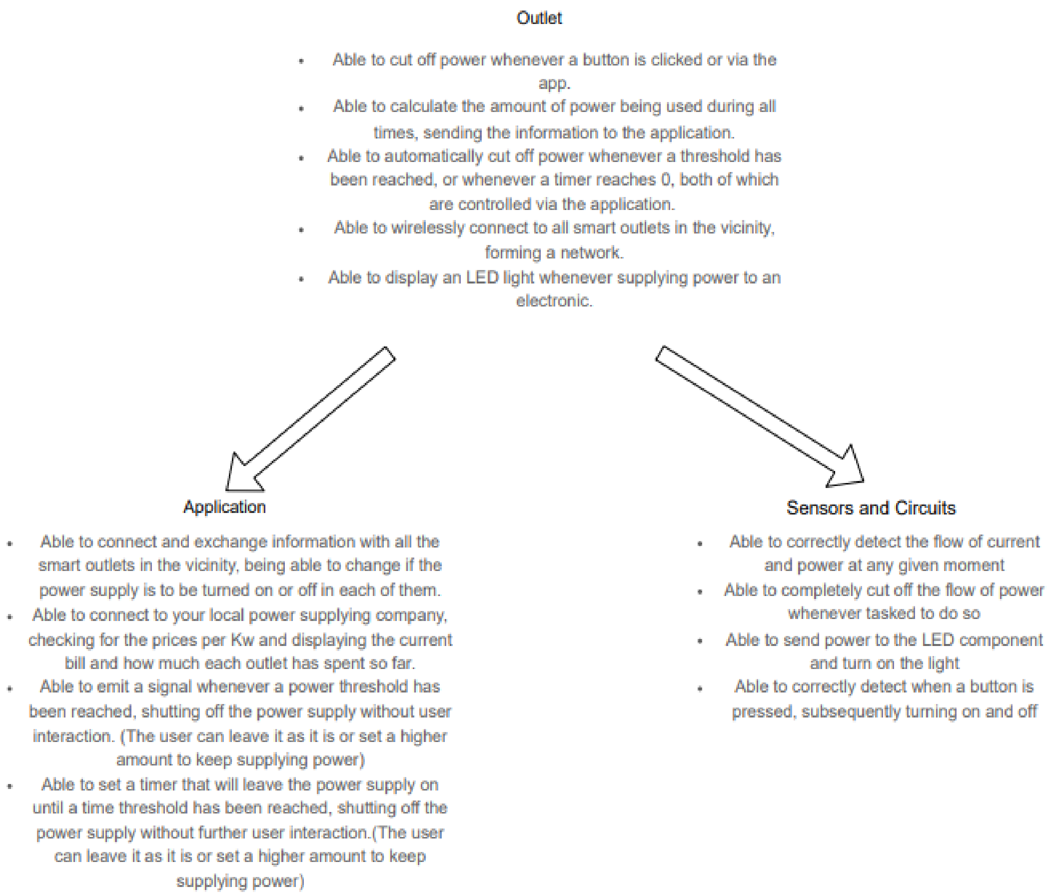
An example of the schematic is shown below. This is a combination of all the individual components of our system. Of course this is not final and is subject to a lot of change once we are able to test the components and confirm what works and what does not work the way we intend to. This is a flexible design that lays the groundworks for what we need to do.

Figure 31. Overall PCB Design Schematic



5.1.4 Block Diagrams

Figure 32. Hardware Design Block Diagram



5.2 Software Design

The design of the software is one of the key components of our project. This includes both the software to control the app, and the firmware to control the microcontroller. Knowing the design of how these should look is crucial to knowing how to quickly develop them. As a result, it is usually a good idea to develop the format the software/firmware should follow before writing any code. In the following sections, we discuss our design and why we picked it, especially with respect to alternative designs. Additionally, we discuss the development tools we plan on using to implement the design. Knowing what tools to use before starting is important as well, since using suboptimal tools can lead to errors and confusion. On the contrary, using well-known and well-supported tools can allow you to have a streamlined workflow.

5.2.1 Design Methodology

For this project, there are two main components working together. First, there is the microcontroller firmware. It takes in basic user inputs as well as the sensor data and sends information to the app. When designing the software, one of the main components to keep in mind is the looping behavior of a microcontroller's firmware. Typically, the firmware consists of a setup function and a looping function. The setup function is primarily used as a way to initialize states and make initial calculations for the code that runs in the loop. The looping function is where the majority of the code runs from, and can be considered the main function of the program.

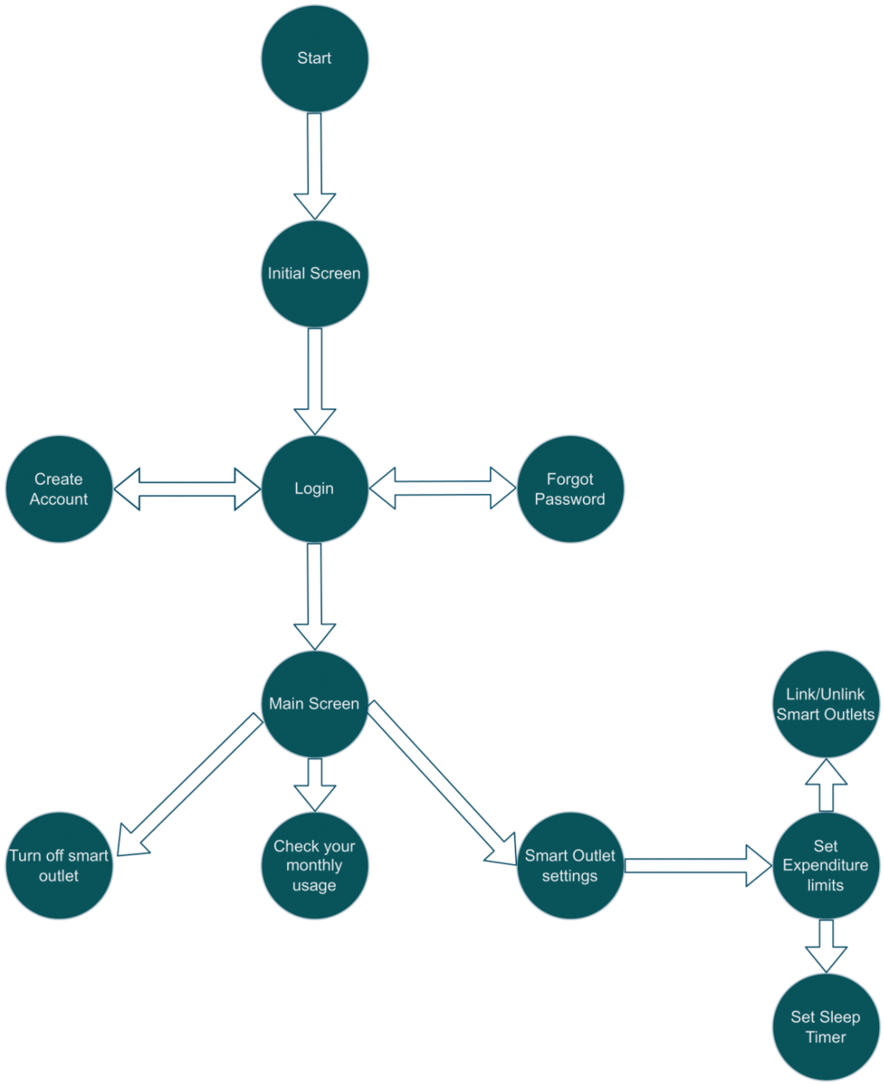
Unlike traditional programs that perform a task then stop, this will perform the same tasks over and over. The repeated tasks usually amount to checking sensors for changes and doing something with that data. For this project, that will consist of reading the voltage of the current sensor, using those values to make meaningful data for the app, then sending that data to the app using a wireless communication protocol. It will also involve using interrupts as well, which are functions that use the state of hardware to interrupt the program and running a specific function, typically called an Interrupt Service Routine (ISR). The user input consisting of the input button will use this, since the user's input is of the utmost priority to the system.

Second, there is the app software. This is software that also needs to continuously update in order to use user-input to perform actions. However, this repetition is a bit more removed from the code that the programmer needs to write. Since developing an app from scratch is remarkably difficult, there are frameworks that provide a foundation that can take care of user-input and program looping for you. In other words, you only need to define behaviors for certain events that can be triggered by the user. This means that writing the code can be considered a bit more simple, though you still have to maintain states and variables throughout the program as you would for a looping firmware program. Since developing our app will be more complex than developing the firmware, we elaborate more on the architecture we want to use in the later app design section.

5.2.1.1 Application Flowchart

Below is an application flowchart, which clearly outlines how a user could navigate through the app. This involves a splash screen, login information, a main screen to show you the most important data and control features about the system, and a settings menu that allows you to edit parameters to further control the outlet system. You first enter the flowchart at the "Start" bubble when you open the app from your device. From there, it describes how you could navigate from one screen to the next.

Figure 33. Software Application Flowchart



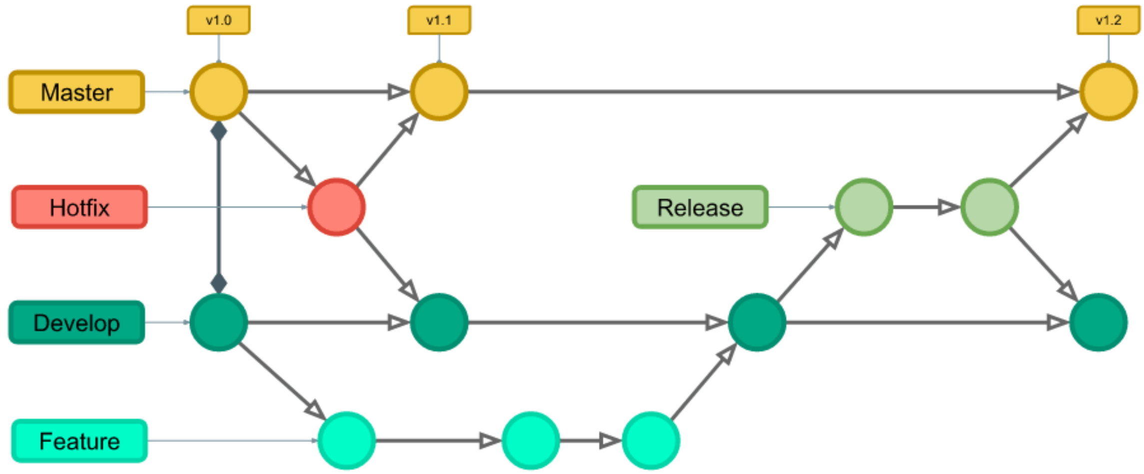
5.2.2 Development Tools

In order for our team to develop this project, we will be using a set of development tools that will allow us to program, manage, and visualize our work. This is extremely important since it allows us to catch any errors or mistakes earlier on, before they get to the hardware and potentially cause a problem. Similarly, this allows us to develop the app faster, since checking code on an emulator of some kind is much faster than deploying the project to a physical device every time we want to test changes. Additionally, some of these tools can allow us to track changes to the code, as well as work on different parts at the same time. This is imperative, since having only one person work on the project's code at a time would be drastically inefficient. Additionally, this applies to the firmware and app development, which need to be worked on simultaneously in order for our team to reach the project deadline. The main tools we plan on using are a version control system, to keep track of our changes, integrated development environments to help develop the code, an emulator to test the app before it is launched to a device, and a text editor to write the code in. We will need two integrated development environments, one for developing the firmware and one for developing the app. Each IDE has a series of tools that makes developing for those specific scenarios easy and straightforward. We also plan on using a good text editor, namely Visual Studio Code. We plan on using this as the editor to write the code because it has a lot of useful features that make development convenient. Additionally, using an app emulator will allow us to catch mistakes early on and develop faster.

5.2.2.1 Version Control System

We plan on using a version control system to help us develop our code, specifically git and github. Git has several helpful features that allow you to manage your code, which will all be very useful for developing the project. One of these features is the ability to save your work in chunks, or commits. This is extremely useful, since it allows you to see how your work grows and changes over time. It also allows you to see your previous work, which is very useful for debugging and tracking changes. Another very useful component of git is the ability to create “branches”, which allows for different flows of code. These different branches start from some existing set of code and allow for changes to be made separate from other branches. Another key mechanic is the ability to “merge” into other branches. This allows the changes of one branch to be applied to another branch. This is very useful since it allows for people to store their changes in a well-known branch that everyone can use. Typically, there is a main branch with all of the most recent code, usually called “main” or “master”, which is useful since it has the most recent changes. Git also has a mechanic for dealing with “merge conflicts”, which is when two people have changes to the same file and git is not sure what to merge. All of these features are very useful, since they are the core mechanics that allow multiple people to work on the same code base at the same time. For example, two people might need to work on different files. Without git, this can be difficult, since copying entire files or even larger folders containing those files can be a hassle. However, when using git, this becomes very easy. Each person can make their own branch, work on their code, then merge their changes back into the main branch. Below is a diagram that helps to explain bit branches in more detail.

Figure 34. Version Control System Diagram



Based on the above diagram and previous descriptions, it is clear to see that using a version control system will be imperative for developing our project with teamwork in mind.

5.2.2.2 Visual Studio Code Text Editor

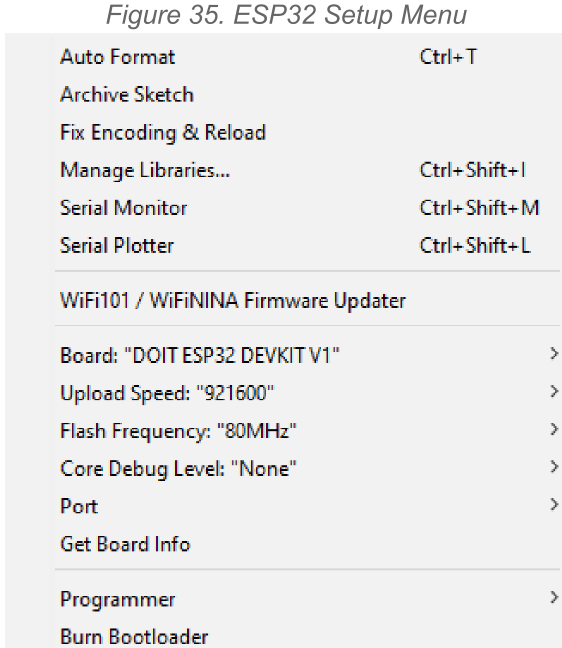
Another tool we plan on using to make development more streamlined is a text editor. Specifically, we will be using Visual Studio Code since it has a lot of community support and the ability to add extensions. These extensions will allow us to configure the text editor to help us develop our project in the best way possible. One of the extensions we will be using is one that checks our code for any errors and highlights them if found. This will likely be specific to the app development, since the extension is much better for the React Native Javascript language, also known as JSX. This extension also allows us to check function calls and definitions from any instance in the code, which makes navigating and debugging issues much easier. It also has an autocomplete plugin, specific to the language and file system we are using. Specifically, it can check for objects and modules we can use and can suggest them to us as we are writing code. Another useful extension we can use is a linter. This uses a file with a set of formatting rules to check your file and ensure that it abides to those rules. Essentially, it can help make our code consistent by showing us where any formatting violations occur. Another unique extension offered is one that allows for you to share your code with other people at the same time. Specifically, you can have someone else work on code on your computer as if they were sitting right next to you. This is great for “pair-programming”, which is when two or more people work together on the same code at the same time. This is especially useful when two or more people need to work on the same file and contribute ideas in real-time.

In addition to the many extensions offered by Visual Studio Code, there are several features available by default that make writing code much easier in the long run. One of these features is autocomplete/suggestions. When you are typing, this feature can make suggestions to you about what it predicts you are trying to type. When you press the tab key, it fills in the first option for what you were typing. However, you can also use the arrow keys to select other predictions, in case the first one is not correct. Another built-in feature is advanced selection and searching. This allows you to navigate your code much easier. For example, if you have multiple instances of the same text and you need to replace all of them with something, you can easily do that with the search feature. If you need to edit multiple things at the same time, you can place extra cursors in the document to do just that. You can also move entire lines of code around as well. There are many more features that you can access with keyboard shortcuts as well, which add to the user experience of the editor. Without listing all of the unique selection/searching options Visual Studio Code has, it is clear to see that it has many more useful features than a standard text editor, each of which make development much easier and more convenient.

5.2.2.3 Arduino Integrated Development Environment (IDE)

In order to interface with the microcontroller, we need an IDE that will allow us to connect to the device and have control over other features related to the device. For this project, we plan on using the Arduino IDE. Not only does it have an enormous amount of support online, it also supports the microcontroller we want to use, the ESP32. This acts as a way to interface with the device as well as upload programs. It can compile and upload your code, as well as give you error messages through a terminal when something is wrong in the code. More importantly though, it has access to edit the way the microcontroller is used. For example, below is the setup menu for the ESP32 microcontroller

Figure 35. ESP32 Setup Menu



This allows you to alter metrics specific to the device. The most important ones to mention are the port and the flash frequency. The port is very important, since it allows you to select the correct device. For example, if you have multiple microcontrollers connected, this allows you to pick the correct one to program and upload to. Additionally, this menu allows you to alter the flash frequency, which corresponds to how fast the microcontroller is processing data, in other words, this is where you can select the clock speed, which is directly correlated with power usage. We will definitely be using this to make our code run at the required speed, allowing for a more power efficient system.

In addition to the microcontroller settings menu, you can also view data in the serial monitor and serial plotter. The serial monitor is a terminal that can display print statement data, allowing for you to analyze what your microcontroller is doing at any given time. This is extremely important for debugging, since it can give you concrete data that can help explain why a bug is happening and how to fix it. Apart from bug detection and tracing, it also serves as a way to confirm that your program is running properly. Below are some diagrams that show what the output could look like.

Figure 36. Serial Monitor Output

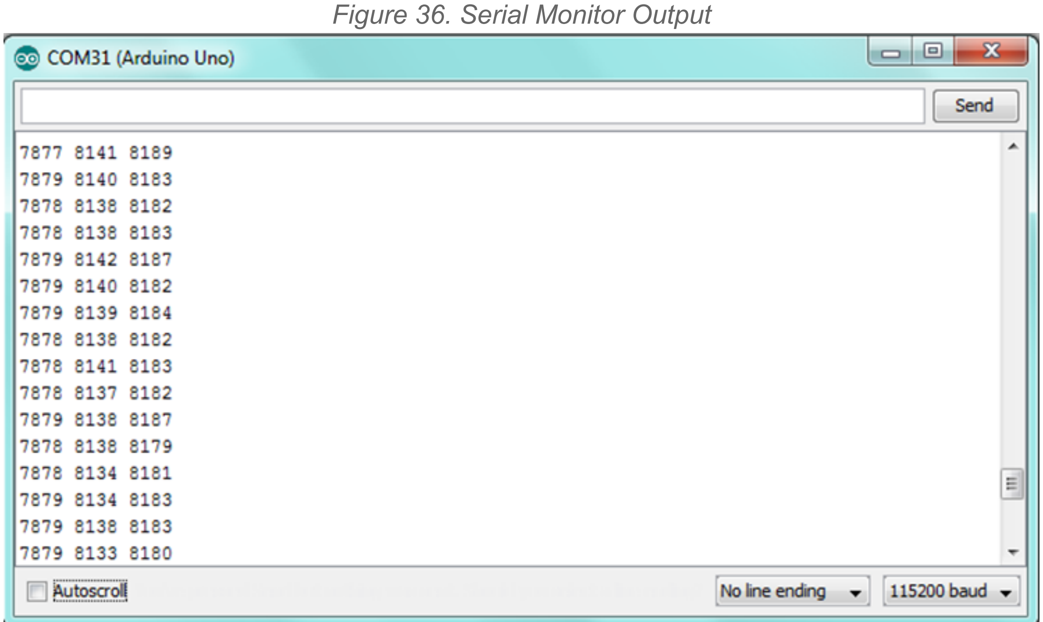
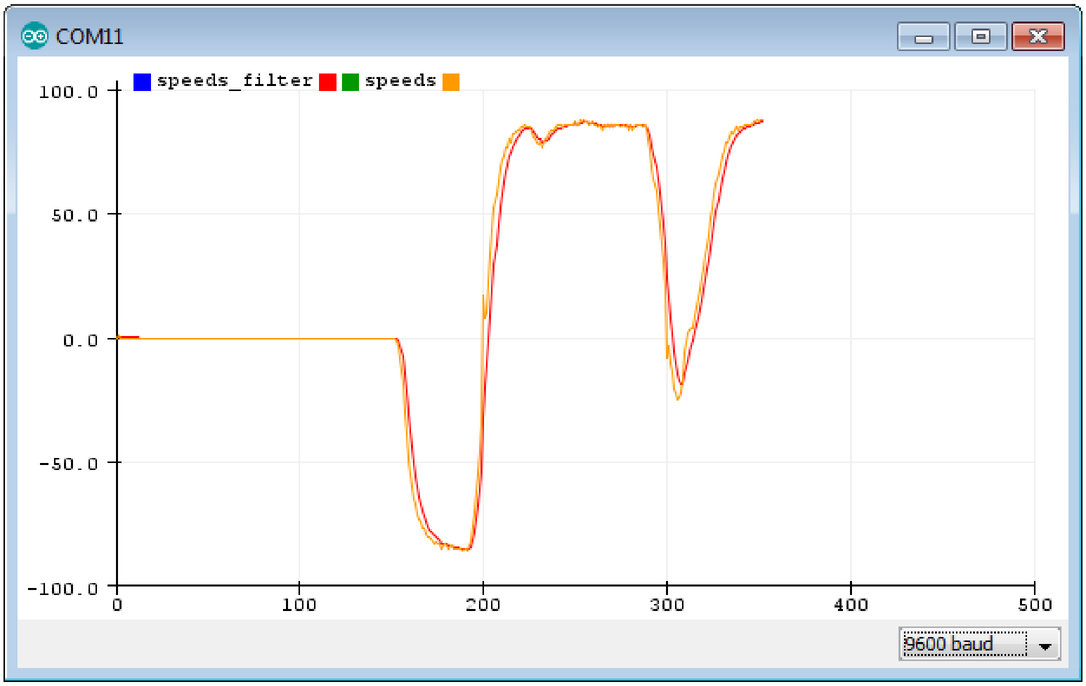


Figure 37. Serial Plotter Output

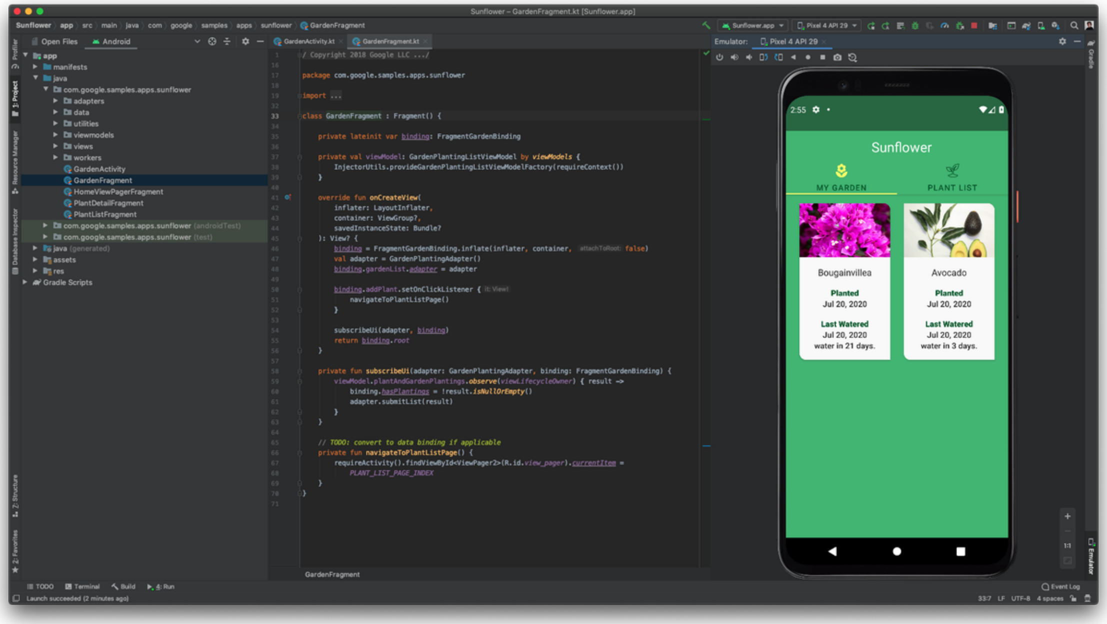


5.2.2.4 Android Studio Emulator

In order to test our app, we want to use an emulator. This is generally really good practice, since it allows for testing an app before it ever gets to physical hardware. Since everything is virtual, uploading the app to the emulator is much faster than uploading to a physical device. As a result, testing the app is much faster this way. The other main benefit of using an emulator is that you can test your app on multiple different virtual devices. For obvious cost reasons, this is much more practical than getting all of the physical devices and testing each one.

The emulator we want to use is Android Studio. Not only does it have a lot of support among the community of developers who use it, but it also supports several devices on the market right now, which makes it very practical for testing our app. Specifically, these devices are android-based, which makes developing on them much easier. Android devices can enter developer mode much easier than ios devices, which makes them a great candidate for testing the product on. Below is an example of what the development environment might look like while running a virtual device, provided by android studio's twitter.

Figure 38. Android Studio Emulator Environment



5.2.3 Sensor Data Processing

In order to send meaningful data to the app from the microcontroller, we first need to make sure the data is as consistent as possible. In order to do this, we need to remove noise from the signal as much as possible. Noise refers to undesirable data introduced from the external environment. In this case, that could be electromagnetic fields (EMFs) emitted from nearby electrical current or undesirable components in the main power signal itself. In order to address noise, there are two steps we can take. One is by using hardware techniques, the other is by using software techniques. We can use hardware to help make the signal less noisy by introducing a low-pass filter. Since we only really want DC voltage being measured, we can apply this type of filter to attempt to remove undesirably high frequencies that made their way into the sensor. Regarding software, we can use averaging techniques to take multiple samples and figure out an approximate value to represent them. This helps to reduce noise and outliers in the data, giving a more consistent set of data.

After processing the raw input of the sensor and converting it to a more usable form, we then want to use it to perform calculations to find additional metrics. One of the metrics we want to send is average power use over time. This involves calculating the power from the sensor data, then using several power values over time to create an average. We can either create an approximate instantaneous average or one that represents power over a more significant period of time. We could also use multiple instantaneous power values over time to calculate a more accurate average of power being used. Additionally, we could use the instantaneous power calculation to generate a sum of power used over time. This is especially useful for finding the percentage of power being used with respect to the threshold set in the app's settings menu. Based on the previous examples and explanations, it is clear to see that we can thoroughly process the data before it ends up on the app's dashboard for the user.

Another important decision to discuss about data processing is where the data processing will occur. It could be done using either the microcontorller's processing power or the app's processing power. We are currently not sure exactly where the processing should occur, though the final decision will depend on which produces less overhead. Our group strongly believes it should take place on the phone's processor, since it is much faster and more robust than the microcontroller's processor. Additionally, this will allow the calculations to have access to constants set by the user, which reduces the need to send them over wireless communication. Testing will allow us to make a more concrete decision about where the processing should occur.

5.2.4 Wireless Data Scheme

The wireless data scheme will consist of two main parts, messages being sent from the app to any given outlet and messages being sent from any given outlet to the app. The below subsections will outline how we plan on implementing each of these two message types into an organized format that can be known across all devices of the system.

5.2.4.1 Shared Message Information

The messages being sent across this system need to always send a few basic pieces of information. The first value will be a delimiter. We will use an uncommon character to help the system differentiate between constant messages being sent across the system and a new message. Since this is a single character, we can represent this as a single byte. The next two values that the messages need to include is a sender id and a receiver id. The size of this value depends on the number of devices in the system. We plan on using one byte to represent this value, since that will allow us to have 256 different devices on the same network, which is more than anyone would ever need. We need a receiver id because the system needs to know where to send the messages to. The sender id works in a similar way, showing where the message came from. In a configuration with a single outlet, this is not very helpful since it adds unnecessary information to the messages. However, if we implement a mesh network into this system, this would show its value much more. Moreover, the sender id would be useful because, if we had a scheme where an outlet is considered the root of the graph, the app would be considered a node of the system. That makes it necessary to specify what messages are from what sender. For more information on the different schemes we would use to implement a mesh network, please see section 8.3.1 Mesh Network.

5.2.4.2 App to Outlet Messages

The contents of the messages being sent to an outlet from the app have certain requirements as well. One of the important values we need to send is the type of message. This would signify if we are sending pairing information, an outlet state change request, or a sensor data request. Given that this is one of three states, we could use a single byte to do this. One of the values we may need to send is the requested state of the outlet. This is only ever on or off, so we could use a single byte to do this. Another value we may need to send is any pairing information. This may require several bytes to work correctly. In the case that the message is a sensor data request, we would only need the message type value to signify that we are requesting sensor data. Consequently, we do not need to send any extra values for this message type. Given the message type value, we would interpret a message from the app in one of three ways. All values being sent up until the message type value will be the same, with subsequent values having different meanings based on the message type. To simplify the code, we would like to use messages with the same length. In order to do this, we will make them as small as they can be based on the constraints of the different possible values we will need to send. The three different configurations are listed below.

Message Type 1, Pairing Mode:

<Delimiter><Sender_ID><Receiver_ID><Message_Type><Pairing_Information>

Message Type 2, Requesting State Change:

<Delimiter><Sender_ID><Receiver_ID><Message_Type><Requested_State>

Message Type 3, Sensor Data Request:

<Delimiter><Sender_ID><Receiver_ID><Message_Type><Extra_Information>

5.2.4.3 Outlet to App Messages

Similarly, the contents of the messages being sent to the app from any outlet have certain requirements as well. One of the important values we need to send is the type of message. This would signify if we are sending pairing information, the current state of the outlet, or any sensor data requested by the app. Given that this is one of three states, we could use a single byte to do this. One of the values we may need to send is the current state of the outlet. This is only ever on or off, so we could use a single byte to do this. Another value we may need to send is any pairing information. This may require several bytes to work correctly. For sensor data, we would need to send average data to the app. This may require either one or two bytes, depending on the data we want to send. Given the message type value, we would interpret a message from the app in one of three ways. All values being sent up until the message type value will be the same, with subsequent values having different meanings based on the message type. To simplify the code, we would like to use messages with the same length. In order to do this, we will make them as small as they can be based on the constraints of the different possible values we will need to send. The three different configurations are listed below.

Message Type 1, Pairing Mode:

<Delimiter><Sender_ID><Receiver_ID><Message_Type><Pairing_Information>

Message Type 2, Current State:

<Delimiter><Sender_ID><Receiver_ID><Message_Type><Current_State>

Message Type 3, Sensor Data:

<Delimiter><Sender_ID><Receiver_ID><Message_Type><Sensor_Data>

5.2.4.4 Messages Working Together

These messages need to be organized to properly work together in the system. The way we will do this is by making the app control everything. Specifically, it will request information from a certain outlet to maintain control of the messages present in the system. Once it requests something from an outlet, the message will move along the network until it reaches the device with the same receiver id. After that, the outlet will send a message back to the app, which will move along the network until it reaches the app. At that point, the information received will be used to display information in the app. However, if we had the app make requests to each outlet all at once, it would cause the network to slow down or even crash. To avoid this, we will have each outlet be accessible in its own screen in the app, constraining the requests and replies to one outlet device.

5.2.4.5 Final Notes

The above sections describe how the app works in terms of a large network. However, this will be the most applicable if we have time to implement a mesh network. In the case that we do not have sufficient time to implement a mesh network, the above communication protocol will still apply to a network with only one outlet.

5.2.5 App Design

As mentioned earlier in this section, the app design is one of the key parts about this project. Knowing the design beforehand will allow us to know how to write the code in a timely manner. Additionally, having a good design will allow us to easily edit or change things later on.

5.2.5.1 App architecture

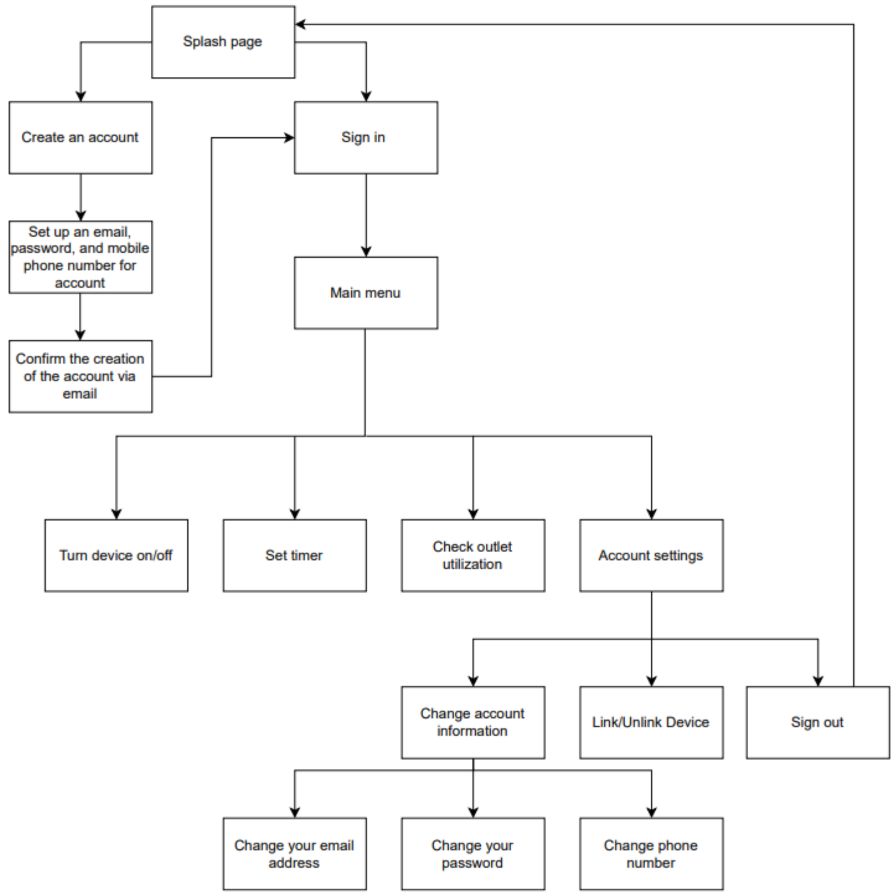
One of the main components about the app design is the app architecture. This defines the core parts the app will use to run, as well as how those parts communicate with each other. There are several different architectures we could use, but we will likely make some modifications to the popular model-view-controller architecture, which is described below.

One of the more popular architectures phone apps and web-apps use to function is the model-view-controller scheme. This scheme has three parts, the model, view, and the controller. The model refers to the data used by the entire system. This can include login credentials, files, profile pictures, and other data. The view refers to the means of displaying information to the user. In this case, that would be the react native framework, since it runs the app's graphics and interface. The controller refers to the part of the project that manages the flow of data to and from the model and view. This connects the system together, allowing it to work as a unified whole. This component is very important, since it allows you to perform actions from the user-interface. In general, this type of architecture is really useful since it allows you to decouple the model and view from the database, which makes the flow of data more clear to understand.

The architecture we plan on using will be based on this, but with some slight modifications. Instead of having a model to store data and call APIs managed by the controller, we will just make these calls from the frontend. Additionally, instead of having a controller, we can use a view component to do that for us. Essentially, the app will all be stored in the view, which makes calls to the backend directly and manages the navigation state of the app. We are able to simplify the MVC architecture to this because we won't be calling any complex APIs that require being managed as much. Additionally, we will only be storing simple sets of data on a remote database, so we don't need to have a model to store complex data locally. As a result of the simple nature of the app, it will be much more straightforward to develop and manage. Additionally, it should run better since there won't be a large web of different parts to manage.

5.2.5.2 Application Flowchart

Figure 39. Application Flowchart



In this diagram, we depicted how the application would work and how each of its features will correlate to each other. One of our main goals for this project is to have an application that will satisfy not only the objectives that it will be set out to do, but also be pleasing to a user. This means that it should have a nice design, making it more appealing for the potential users of the outlet.

Additionally, we will be greatly focusing on making sure that this application will be safe and bug-free. Most of the information being given by the user of the outlet is of extreme secrecy and would cause harm if an unauthorized third party got ahold of our project's database. Our team is going to be very focused on the cybersecurity side of the software, making sure that there's no way someone could just collect that information. A product considered "safe" attracts more attention in the eyes of a potential user of our outlet, and will be given its intended attention.

5.2.5.3 Testing and Simulation

In order to make sure our project is working properly, we need a way to test each of the primary components. With these tests, we can constantly check for errors and make sure that our code is functioning properly. There are two parts of the system we need to test, firmware tests and software tests. These systems are physically separate from one another, so testing must be done separately as well.

5.2.5.3.1 Firmware testing

For the firmware, we need a way to test the code to make sure it is behaving properly. Setting up testing programs that run through different scenarios is a bit difficult for microcontroller testing, so we will instead use a physical testing environment in combination with testing programs. Specifically, we will use test programs to read simple amounts of data and compare that data to the values read by a high-precision oscilloscope. An oscilloscope is very accurate, so we can use that to compare expected behaviors from the test programs, which will help us evaluate the accuracy of the different parts of our system. Moreover, we can test each of the individual components before they are combined into a unified system.

Regarding the current sensor, our testing will involve using a variable power supply to provide different voltage and current through the current sensor, giving us a wide range of values that we can test. For this scenario, we want to connect the output of the current sensor and the input voltage to an oscilloscope and verify that the conversion from one to the other is consistent. This will allow us to make sure the output of the sensor itself is what we are expecting from the analog to digital converter, or ADC.

That being said, the next component we want to test is the ADC. Specifically, we want to connect the variable power supply to the inputs of a differential channel and provide a range of values that would be expected from the current sensor. In addition to this, we want to check the value of the reading registered from the sensor. To do that, we will write a basic program that reads the input and prints the output to the serial monitor. This will allow us to see the actual values the sensor is producing based on the given input. Additionally, the Arduino IDE comes with another feature called the serial plotter. This will take the numbers from the serial monitor and graph them. This will be very useful for us to visually see how the input voltage gets translated to a digital form. Moreover, if we connect the input voltage to an oscilloscope, we can get a graph of the input as well. This will allow us to compare both graphs, which will be very useful to determine any error in the sensor data.

The next component we want to test is the relay. Specifically, we want to test the circuit that powers the relay, where the state of the relay is the testable metric. Relays are very simple modules, they have an electromagnet that pulls a contact from one position to another. This allows it to switch a common terminal from one output to another. Fortunately, this means that powering an electromagnet can change the state of a switch, where the electromagnet and switch are completely electrically isolated from each other. However, its basic construction is not without drawbacks. When powered, an electromagnet builds up an electromagnet field and stores energy there. When power is removed, that magnetic field collapses and induces a relatively high current back into the connected circuit. This is often called a flyback current, which usually manifests as a negative voltage. This would damage the microcontroller if the GPIO was directly attached to the input of the electromagnet, so we need to somehow decouple it. The most common way to remove this is to use a flyback diode, which only allows current to flow into the magnet, and dissipates flyback current safely. This means that we need additional circuitry to drive the electromagnet in the relay properly, which is what we need to test. In order to test this sub-system, we need to make the circuitry to drive the diode then provide a unit step input to the system with a function generator. If we connect both the function generator input and relay output to an oscilloscope, we can compare the results and verify that it works properly. We will also be able to test for latency and other metrics and possibly improve them.

The last main component we need to test is wireless communication. The tests for this will be fairly simple, since the ESP32 calculates various metrics for us. Regarding the wireless connection, we essentially want to verify that it works and that latency is relatively low. In order to do this, we can measure a round-trip time for a signal to be sent from the ESP32 to the phone and be retransmitted back to the ESP32. This will allow us to figure out how fast our wireless communication is. Additionally, it will help ensure that the communication is working properly. Testing this will give us insight on how we could potentially reduce latency and increase the response time of the communication.

Based on the explanations above, testing all of the individual components of the hardware system will be very useful for several reasons. One of those reasons is that we will be able to prove the concept of how the system will work, one step at a time. Additionally, it will allow us to see any error in the expected result, which can tell us how to improve the system. Testing will likely be done several times to ensure that everything is working properly. It should almost always follow the general form of providing an input and comparing it to the output to find the error, which is useful for the reasons specified above.

5.2.5.3.2 Software testing

The software testing will follow the same form as the hardware testing, which is providing an input and testing for an expected output. However, it will be a bit different since we only expect one specific output for a given input. For example, if the “settings menu” button was pressed, we would expect it to go to the settings page. If we pressed the “toggle power” button, we would expect to receive a message from the microcontroller verifying that the state has been changed, which would update the state in the app. These are just a few examples, but essentially each feature should be tested in some capacity. For software this can be done in one of two ways. One way is by manually using each feature in each possible way it can be used, then recording the behavior it produces. However, just as it sounds, this is not very efficient and would take a lot of time to do. The other way to do testing is by automating it into a test file. Essentially, this will make a virtual user input for the device and will have the ability to interact with the device through software. Since it will all be through software, we can easily compare the virtual user actions to the states of the app. This will allow us to determine if each functionality is either passing or failing, which can tell us where to find and fix errors in the code. This should be much easier to do, since we can write all of the tests in software instead of having to configure hardware for each component we want to test. We plan on performing these tests through a testing library provided by react native, which is the foundation for the app. Since testing is integrated into the framework we are using to develop the app, it should streamline the process for making tests.

5.3 Summary after project completion

After concluding our project, we can confirm that some of our fears came to be true. One of our major constraints was size, given the fact that our project had many essential parts, we had to build a 3D printed case in order to fit all of them in, giving us an extra roadblock during the design of it. When it comes to standards, we always kept them in mind, installing safety nets for the user's protection if something goes awry.

6.0 Project Prototype Testing Plan

The following sections will detail how we plan to test the hardware and software components of our prototype. Specifically we will cover the wifi connectivity of the ESP32 microcontroller, the accuracy of the analog to digital converter, the accuracy of the current sensor, the speed and consistency of the relay, and the ac to dc converter. We must confirm that all of these components work separately before even having a chance to combine them into a larger design. This is imperative to our success as a team and therefore should have consistent results in whatever components we are using. As for software we will cover some of the expected outputs in the testing in response to our inputs within our IDEs. Although the software is a future component of our project, it is an important aspect of it and we can try to ensure that we are able to communicate with our devices through wifi if needed.

6.1 Hardware Testing Environment

Our hardware will be tested using a multimeter, digital oscilloscope, light emitting diodes, leads, and breadboards. We will also utilize waveform generators and power supplies in order to simulate a load for our components to use as power or to measure certain attributes of their design. Lastly if needed, we will use the spectrum analyzer to help us troubleshoot any signal errors if needed. Doing these tests will enhance our ability to design the PCB correctly and update it if needed regarding any sort of errors or points of contention.

As the wide variety of our components are available as stand alone devices, we are able to test them individually and them in congruence with the other devices. This ensures that we are able to troubleshoot them in a basic state before utilizing them in a bigger picture project. This is especially useful for the ESP32 as we are able to use a hot plate and desolder all the individual components of our testing ESP32 and use them in our prototype if needed. One we ensure that each component works well separately and together in the most extreme parameters we are able to do by the testing devices given to us, we will then be able to test it using a wall plug. I say extreme parameters as the function generators only are able to go to a max delta of 20 volts which pales in comparison to what will be coming out of the wall.

6.1.1 Current Sensor Testing

Below are 2 inputs and outputs from our current transformer sensor. For both figures, the yellow line is the input and the green line is the output. The first figure is when the wire of the input is pushed very tightly against the solenoid of the current sensor and the output is produced. The second figure is when the wire is loose and not tightly pushed against the solenoid of the current sensor. What is made apparent from our testing here is that when we create our outlet we need to make sure that the wire that provides the power is tightly bound to the solenoid of the current sensor otherwise an inaccurate output may be displayed. What also can be seen from the graphs is that the voltage that is input is not what is output. This is normal with the current sensor and the correct information will be translated to the app when they are connected.

Figure 40.Current Sensor Testing Tight

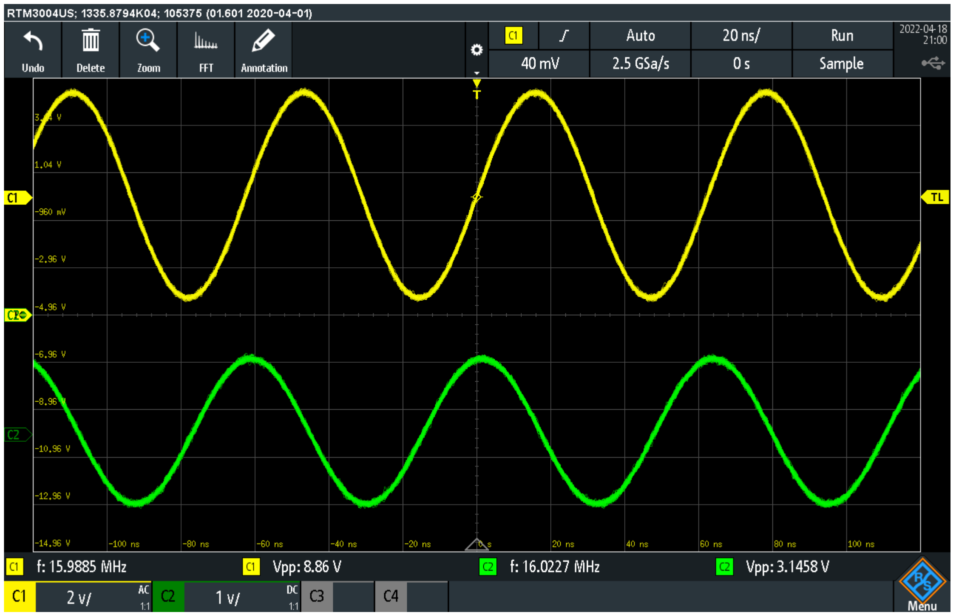
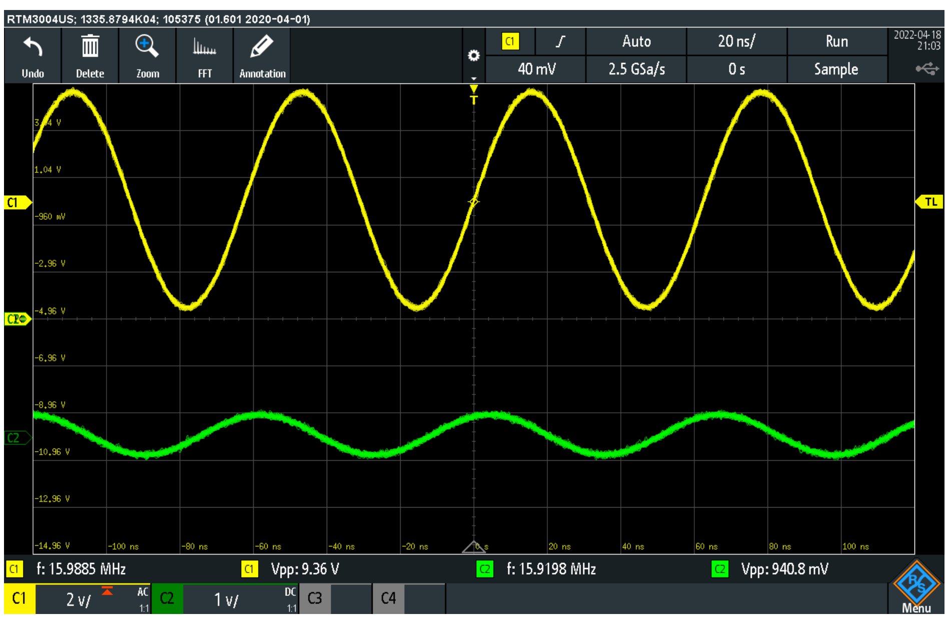


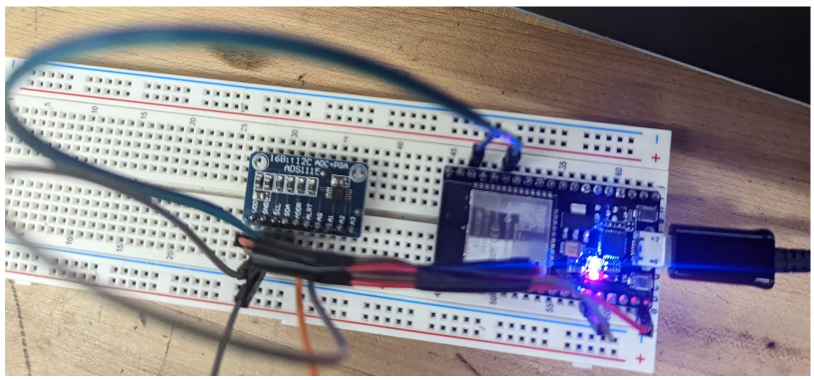
Figure 41.Current Sensor Testing Loose



6.1.2 ADC Testing

For the analog to digital converter testing, our goal was to see how accurate the measurement was based on an analog signal. For this we connected the ADS1115 to the ESP32. The ESP32 was used as a communication between the board. For testing the voltmeter we measured the delta between the voltage reading of the two input pins. This is known as putting the adafruit in differential mode. The output of this board was sent to the ESP32 and read on the arduino compiler. The setup of this test is shown below.

Figure 42.ADC Testing



For the adafruit measurements, the ADS 1115 has a power supply voltage max of 7 Volts. Although we would prefer not to use this amount of voltage we measured the accuracy between both 5 Volts and 7 Volts. This allowed us to see the differences between the 2 voltage levels as the max reading output is dependent on the voltage input. Considering what devices might be plugged into the outlet will determine our decision of voltage input as well as what other devices require to be powered. Below are our measurements between 5 volts and 7 volts and different voltage measurements or different voltage differentials.

Table 13. ADC Readings

	5 Volts Vdd	7 Volts Vdd
3 Volt Differential	3.0012	3.002
4 Volt Differential	4.0001	4.0004
4.5 Volt Differential	4.4998	4.5006
5 Volt Differential	4.98	5.003
5.5 Volt Differential	5.23	5.5001
6 Volt Differential	5.4	5.9993
6.5 Volt Differential	5.5	6.4997

7 Volt Differential	N/a	6.989
7.5 Volt Differential	N/a	7.245

During testing of the adc we realized that increasing the sampling frequency would cause our reading in the oscilloscope to be of higher quality. By this I mean that there would be less faults in the waveform when compared to a higher sampled waveform. The lower sampled waveform would have lumps or curves rather than a straight line. For us this does not cause that much difference in final results but is something we will have to take into account when creating the final system. Below is a waveform of the actual signal and the measured signal taken at higher sampling frequency. You can see how consistent the waveform is throughout and is very similar to the actual waveform. The line does slightly move throughout but when compared to other signals this is very minimal.

Figure 43.Analog Signal

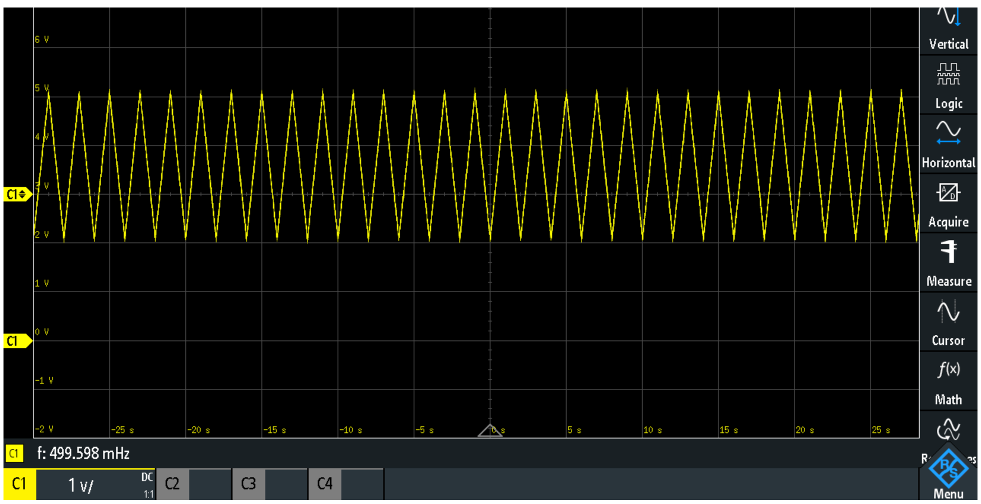
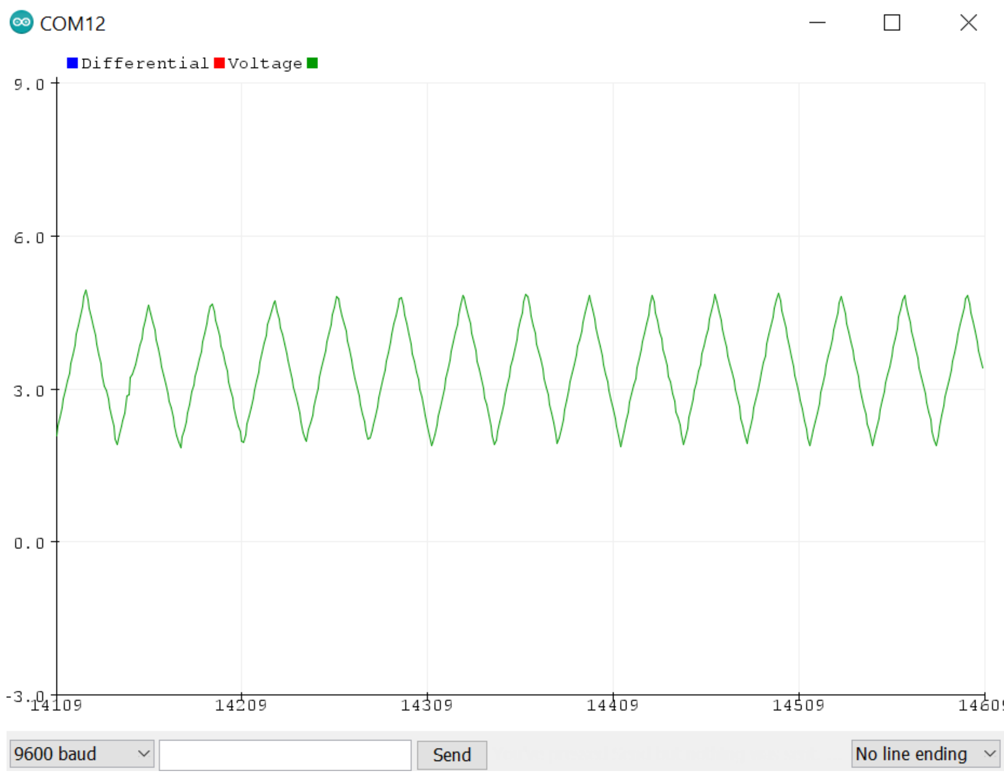


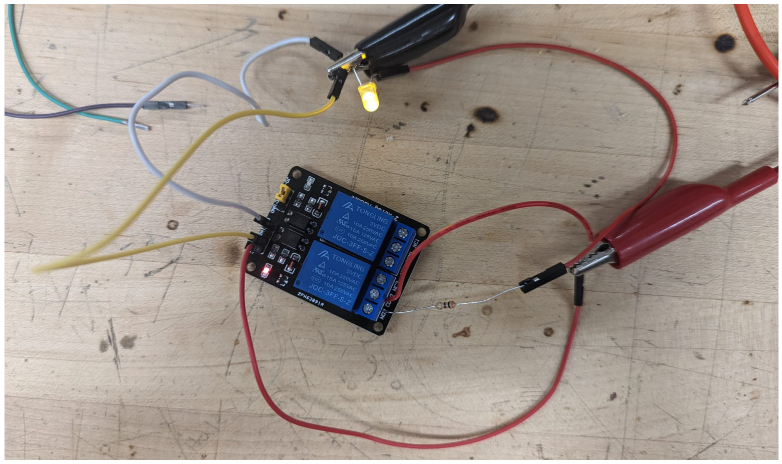
Figure 44.Analog with Analog to Digital



6.1.3 Relay Circuit Testing

From our research most boards with an optocoupler utilize negative logic in order for the relay to function. This board is no different as the input channel must be grounded for any change to occur within the NC and NO connections. We connected a closed circuit of a simple led and when the input pin was grounded, the circuit was completed and the light emitting diode turned on. Below is the simple layout used to test the board. In the picture you can see both the led on the board turn on as well as the external led of our circuit. The reason we tested the relay using the external LED is that during our research we found an example of the board LED turning on but the circuit not working as intended. In other words the relay was faulty. In order to assure that this was not an issue with our relay we tested both the external and board connections.

Figure 45.Relay Testing Circuit Example



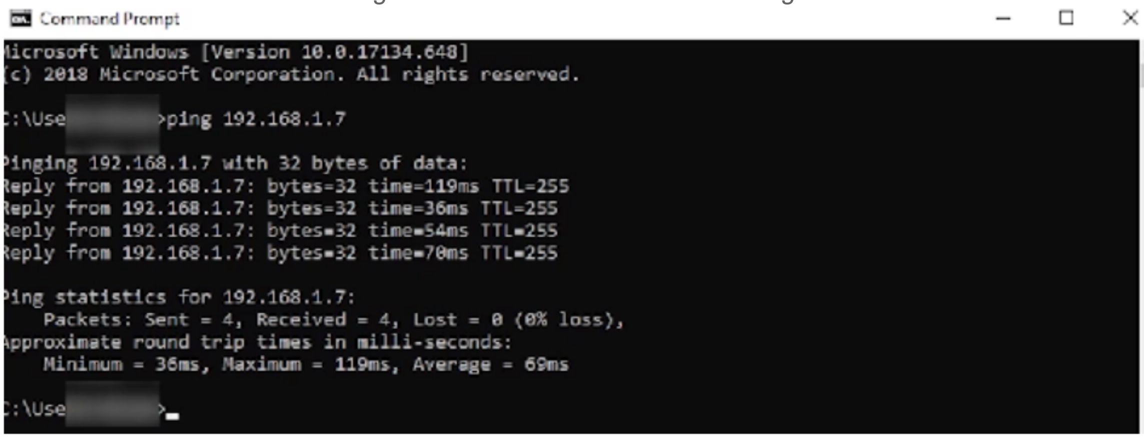
We tested the board multiple times as well as utilized a slow 1Hz square wave and the relay functioned every time as expected. We repeated this test at a higher 2Hz square wave and confirmed that it still worked as expected. The board was not tested at a higher frequency as we felt it was unnecessary given its use case. The relay was tested manually very quickly by physically grounding the input and met all expectations.

6.1.4 ESP32 Wireless Communication Testing

In order to ensure that our system would be able to work remotely, we needed to test the wireless communication of the ESP32. In order to do this we connected the ESP32 to wifi. This was done through a library in Arduino. The board needed to know the name of the wifi network as well as the password. Furthermore, in order to not allow the ESP32 to attempt to connect to the wifi infinitely we gave it a timeout of 20 seconds in order to make sure that it would stop if there were any errors. The ESP32 has two modes of operation within the wifi chip: station mode and access point mode. As we are connecting to an existing wifi network, we utilized station mode. We are utilizing station mode as we expect the outlet to automatically update its power usage to the internet rather than wait for the user to connect to it and get updates when it has worked with a user.

When connected, the console would print out connected followed by the IP address of the board. We utilized the command prompt of windows to confirm that the board was connected by pinging it. Below is the response we received.

Figure 46. ESP32 Connection Testing



From here we would be able to transfer information through a communication protocol such as SPI or UART and this would help us with the transmission of the information regarding the power usage of the outlet.

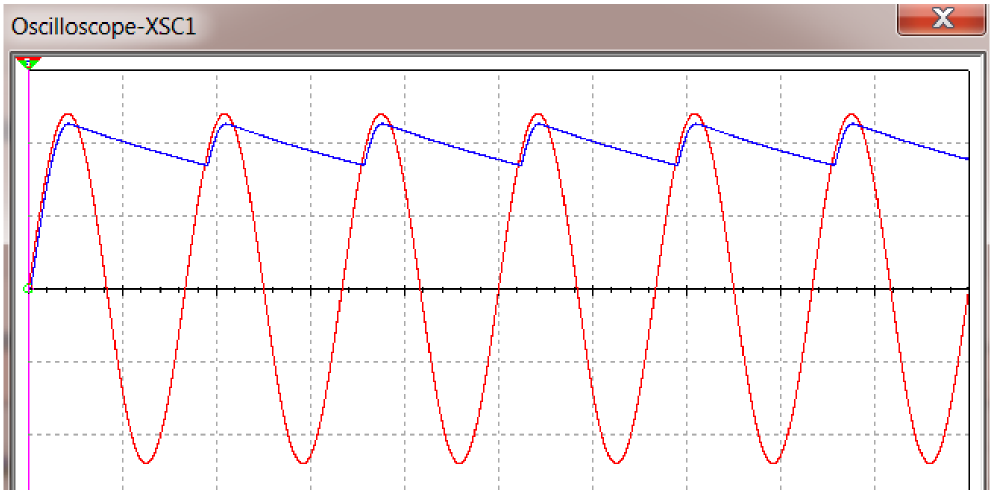
For the first test we would send a string of data regarding the power usage from ADS1115 through the ESP32 and would read it from the console. For the second test we would want to test the GET and POST commands. We would create a POST command to write into and API a JSON string. We would POST the successful command from a created string. Finally we would try to GET the POST command from before. This would allow us to ensure that both commands were working. This would be done if we needed to send some sort of command from the ESP32 based on the prior information

6.1.5 AC/DC converter testing

For the AC/DC converter, we would utilize the function generator and the oscilloscope in order to confirm the success of the converter. Although the module we utilize has a stepdown transformer, we would be able to measure the capabilities with a function generator despite being outside the minimum recommended range. We would test the consistency of the DC output and make sure that the peaks are minimal in order to ensure that our device is able to rely consistently on the ac/dc converter.

Below is a rough estimate of what we are expecting. It is created using a full bridge rectifier circuit. One of the possible methods we researched in order to convert ac to dc manually. The red is the input ac signal and the blue is the somewhat dc value. We would expect our dc signal coming from our real converter to be more stable and have less falls.

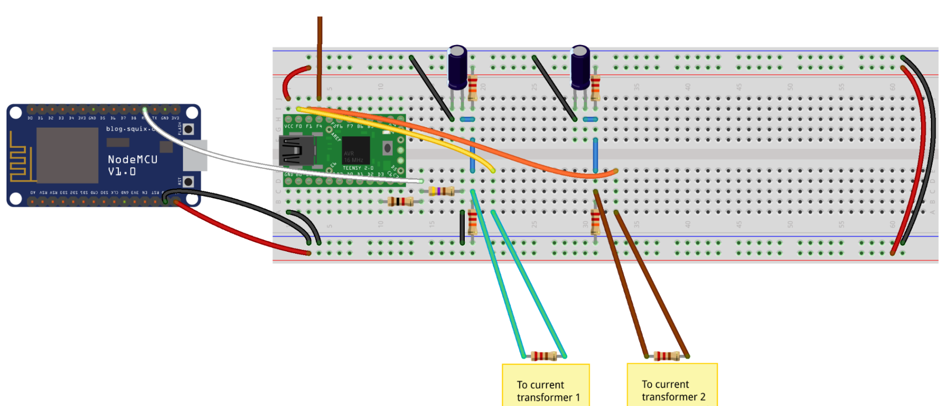
Figure 47.AC to DC Reference Waveform



6.1.6 Testboard Example

Testing all the components individually is great but our project is a combined components design and therefore everything must be tested in conjunction to what is stated above. Below is an example of how this might look. The current sensor would be measuring the hot wire of an ac signal. The current would be transformed over a burden resistor of which we would be able to read the voltage going through it using the analog to digital converter or the ADS 115. This information would be read by the esp32 and transmitted wirelessly to a database in which we could communicate with. After a certain amount of time we would time the relay connection with the ESP32 in order to make sure that all the components are able to work in tangent with each other. Of course this is just a simple explanation of how everything should work together and as more time and effort gets put into the final hardware testing and pcb design we will be able to more accurately describe in detail the workings of all the components. Below is a testing example of the components working together.

Figure 48. Breadboard Prototype Test



6.2 Summary after project completion

One of the most important parts of our project was the project testing. After having built the project, it came down to testing the outlet. In order to check the voltage being emitted by the project, we used some electrical devices and if they were working and pulling the right voltage at any moment. When it comes to application testing, we would check if the numbers being shown on the app consisted with the ones in the multimeter. When all those numbers checked out, we knew we were done.

7.0 Administrative

This section is meant to hold more administrative information rather than technical information. Specifically this includes an estimated budget, estimated timeline, and stretch goals for the project that we would like to implement if possible. This section is important since it easily outlines what parts we plan on using and gives us an idea of if we are on track to finish the project or not.

7.1 Estimated Budget and Financing

Table 14. Budgeting and Financing Estimations

Part Link	Name	Multiple of Order	Unit Price	Projected Amount	Vendor	Alt	Notes
Link	ESP32 Microcontroller	1	$10.99	$10.99	Amazon		
Link	Standard Outlet	1	$5.89	$5.89	Home Depot		
Link	AC Current Transformer Sensor	1	$10.99	$10.99	Amazon		
Link	Relay module	1	$6.99	$6.99	Amazon	Link	Parts will be moved to custom PCB
Link	Custom PCB	1	$10.00	$10.00	JLCPCB		Includes 2 week shipping
Link	External ADC	1	$12.76	$12.76	Amazon		Needed for higher precision and 1V max AC differential reading
Link	120V AC to 5V DC converter	1	$11.50	$11.50	Amazon		
		1	$30.00	$30.00			This should ideally account for any extra parts we need to order
			Total Cost	$99.12			

This bill of materials outlines a general idea for what we need to purchase for our project. This is subject to change since we may learn that we need more during the research phase of the project, which is outlined in the project timeline below. Specifically, we will likely need to add more components to the list based on the PCB design. This is because we need to figure out the supporting circuitry needed to make the PCB function properly, which will be found out during the research component of the project. Additionally, the above devices primarily serve as more basic components that we can use to test the overall design of the system. With that being said, we will also likely need to add more compact versions of the components above in order to fit them all in an enclosure.

7.2 Estimated Timeline

Below is the estimated timeline for our project. This will be edited as we progress with our project, becoming more accurate with each edit. Ideally, we would like for our timeline to shift up, allowing for more time to implement and develop the project.

Table 15. Estimated Timelines

Weeks	Milestones
1	Come up with project idea
2 - 3	Create initial project documentation, requirements, and potential implementations. Discuss roles and tasks for implementing the project and dividing labor.
3 - 6	Research: Which microcontroller we need to use. How we can convert 120V AC to 5V DC in a small form factor for the microcontroller. How we can accurately use a split-core current sensor to read voltage. How to interpret the sensor data in a meaningful way for the user. What scheme we should use for developing our app. What our app should look like. How the app connects to the microcontroller.
6 - 7	Work on the circuit design
8 - 11	Write documentation to explain researched information
12	Delegate the build tasks among each project member
End of Spring Semester	
13 - 14	Hardware: Order all of the materials outlined in the current bill of materials. Work on PCB design App: Start by importing boilerplate.
15 - 17	Hardware: Receive components and start assembling the system. Microcontroller: Begin writing code. App: Make several screens and connect them with simple buttons.

18 - 20	Hardware: Begin testing and make any adjustments to the circuit. Microcontroller: Work on wireless connection to the app (bi-directional). App: Populate each page with some core features, such as a dashboard, an on/off button, and a settings page to set thresholds and connect to other devices. Begin testing wireless connection.
21 - 23	Hardware: Begin modeling enclosure for all of the hardware, 3d print prototype versions. Microcontroller: Work on sending/receiving the data over the wireless connection. App: Finish making wireless connection, connect received data to visualization dashboard, connect button to transmitting data.
24 -26	Fine-tune the system to make sure it works properly during the presentation.
27	Present

Although it is very likely to change, this is the timeline we have set for our project, including every major function we plan to implement. Following this timeline is not something we should follow strictly since the time to implement certain parts of the project may vary. This may be due to needing to perform additional research or wait for parts to arrive in the mail. However, if a set of goals is completed ahead of schedule, it is important for us to proactively advance to the next set as soon as possible. This is especially important since the next set of goals may not be as simple to accomplish as we first imagined.

Additionally, it is important for our group to order parts far in advance in order to remove as much variability in the timeline as possible. A delay in a part's delivery could potentially be a major hindrance in our project, even causing it to fail in the case that the delayed part is critical to the device's functionality. Moreover, determining each of the parts we need to get and which companies we should purchase them from should be one of our top priorities. This will help ensure that our teammates will not have to make up for lost time in the case that some of these parts do not arrive on schedule.

7.3 Stretch Goals

Through our project, we have been discussing possible features that may be implemented in the case that we first complete all the necessary aspects of our project. This section is to allow for explaining these ideas such that we have a guide to follow if we are able to pursue implementing them. In each of the following subsections, we introduce each of the concepts we have in mind and some possible ways we could implement them. This will allow us to have a guide in the case that we have the time to incorporate them into the final product.

7.3.1 Mesh Network

One of the more ambitious features we would want to implement would be a mesh network among various smart outlets that are both registered to and within the vicinity of the user. Essentially, we mean to interconnect several outlets as nodes, using one another to communicate with and send information to the companion application. The purpose of this is to allow the app to conveniently control and monitor several smart outlet devices in a user's home. This is highly practical, since many users using this system would likely want to monitor more than just one outlet. Ideally, we would have one node as the master/root node, which has no parent node and can communicate with the rest of the nodes. The other nodes will be able to send and receive messages to and from one another in order to allow the message to make their way to and from the root node. This is similar to how internet switches can propagate information amongst their neighbors in order to get it from one place to another. For the sake of simplicity, we will explain this in terms of graph theory since it is a great model for describing nodes and edges, which is exactly how our network would be set up. When referring to nodes, the root is an important one to consider since it establishes how the rest of the nodes connect to it and communicate with it. As a result, we will explain the following options with that as the main focus. As far as implementation goes, we have discussed a few main ways, which are further explained in the sections below.

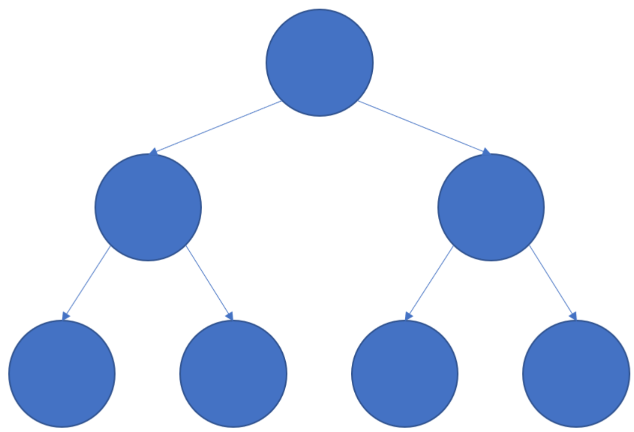
7.3.1.1 Graph Designs

As mentioned before, the root is an important deciding factor on how the communication scheme is built since it defines where the rest of the nodes will send their information to communicate with the rest of the system. There are two primary ways that the root node can help define the overall graph, which is explained in the next sections.

7.3.1.1.1 Binary Tree

One way to implement this is to have a binary tree structure. This means that each node can have only up to two child nodes connected to it. In our case, we would likely want to use a perfect binary tree. This is a binary tree where all of the terminal nodes are on the same level. This makes the overall graph more filled out and better approximates faster logarithmic access times. Due to the faster access times, this architecture would be beneficial if we want to use a large number of devices. Additionally, it makes sending information from the root to any other node fast and easy. However, one of the drawbacks of this scheme is that it would require each node to store information about its child and parent nodes. That is relatively straightforward and requires little extra data, though we would need to account for possibly having large trees. This means that we would need a way for messages to propagate from nodes on the lowest level of the tree back to the root, which would make implementation much more difficult and lengthy in code.

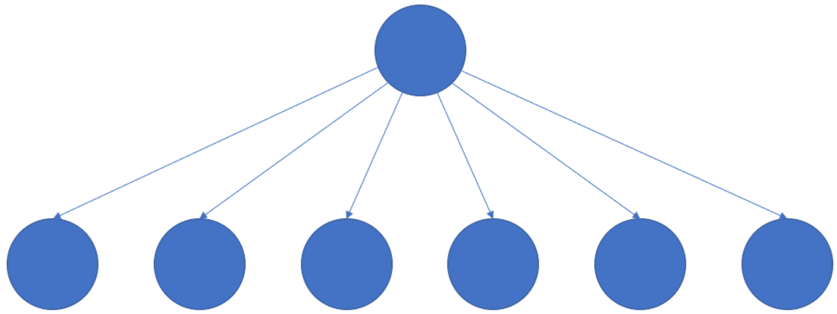
Figure 49. Basic Binary Tree



7.3.1.1.2 Non-Binary Tree with Height 1

One of the other ways is to have a non-ninary tree structure with a height of one. This means that the root node can have several children, not limited to only up to two nodes like the aforementioned binary tree structure. The root node connects to each of the additional child nodes, which are also terminal nodes. This would give the graph a height of one. This would allow for streamlined communication to the root and subsequent nodes, as well as making the process of adding additional outlets to the system easier. Regarding the programming implementation, it would be fairly straightforward since each node only needs to store information about one parent and one child node. However, it would require the root node to store information about each outlet in the system, which would cause memory usage to increase quickly. Moreover, this would cause a lot of traffic to go through the root node. Given that the root node would be an outlet powered by a microcontroller, this added traffic may have negative effects on that outlet's performance.

Figure 50. Binary Tree Height One



7.3.1.2 Locality of The System

The system will use Wi-Fi protocols to communicate with one another, though it can do this in two main ways. It can use Wi-Fi to establish peer-to-peer connections locally or over the internet. This essentially establishes whether you will be able to connect to the system from outside a close proximity or not. These options are explained in further detail below.

7.3.1.2.1 Local Peer-To-Peer

One of the ways mentioned above involves using a local peer-to-peer architecture. This means that the root will be able to communicate with the nodes directly using Wi-Fi communication protocols, but not actually connecting to the internet. Essentially, the connection would be localized to the user's home and only be available in relatively close proximity. For this case, two devices could be the root of the graph, either the app itself or one of the outlets. If the app was the root, it would establish connections to each child node as explained in the "Graph Designs" section. If one of the outlets was the root of the graph, we would need to establish the app as one of the nodes as well. However, the app child node would need to be special, since it will always be the destination of outlet child node messages. The benefit of this system is that all of the devices are local, so message sending latency should be considerably low.

7.3.1.2.2 Internet Peer-To-Peer

Another way we could connect the devices is by using Wi-Fi to connect to the internet, allowing for peer-to-peer connections. The difference between this and the previous option is that, with this scheme, we could connect to and manage an outlet system from anywhere in the world. Similar to as listed above, we could establish the internet peer-to-peer in one of two ways. One of these ways would have both the root node and the child nodes connected to the internet individually. With this configuration, it would make sense to make the app the root node and use the previously mentioned non-binary tree architecture with a height of one. This would essentially connect the app to each node with its own individual connection. One of the drawbacks of this is that it requires a lot of connections to the internet, which typically increases message sending latency.

The other way that we could do this would involve using both the internet and local connections in a sort of hybrid fashion. Since this uses concepts from the previous two sections, it has been put into its own section below.

7.3.1.2.3 Hybrid Design

The last alternative way that we could connect the devices is through a hybrid architecture. This would involve using the internet to connect to a single outlet from anywhere in the world. That would also establish one of the outlets as the root node, which would then be able to send and receive messages from the rest of the other outlets locally. The app would be considered a child node as well, but with special properties. It would need to always be the destination of information from the outlet nodes. The benefit of this design is that we can increase local connections and reduce internet connections, which should overall increase latency while allowing for access all across the world.

7.3.1.3 Deciding on an Architecture

As of right now, we are unable to fully know what will be the best option for our system without doing more testing and getting more information. If we have the time to fully pursue adding this feature to the system, we will need to analyze the pros and cons of each of these communication architectures with more metrics involved. Having more metrics and measurements recorded would allow us to make a fully educated decision on which scheme to use with data to back up our claims.

7.3.2 Increased Calculation Accuracy

Initially we will likely have more imprecise readings in order to get a working product faster. However, if we have the time to do so, we would like to increase the accuracy of our readings as much as possible. There are two main types of accuracy optimizations that we would want to make. One of them would be the ability to calculate averages more accurately based on the data we read from the sensor. This could involve more complex algorithms, which will take more time to research. The other optimization would be increasing the accuracy of the readings themselves, before they even get to the averaging algorithms. This would increase the average accuracy even more, allowing for more accurate data readings. These improvements both contribute to having better data averages, which is where all of the subsequent power calculations will come from. So naturally, if we improve the averages, everything becomes more accurate. These optimizations would help a user have a much better idea on exactly how much current and power they are utilizing every month.

7.3.3 Multi-platform compatibility

Another stretch goal would be to make our application compatible with both Android and iPhones. Due to ease of development, our main goal is to release an application for Android devices only. However, only having Android compatibility is very limiting since a major portion of the general populace owns an Apple device.

Our current development environment, which is explained in the “Development Tools” section, only supports easily developing for android. It can port the app code over to apple devices, but it is extremely difficult. In fact, several app development environments only support making an app for one of those two main platforms. In order to make it compatible with apple devices, we would need to either code the entire app again in a platform like Xcode or work around many hurdles through react native. Since none of us have worked with developing apps for iOS, this would likely require months of research and time. If we have the time, we would like to implement this. To be realistic though, we will not likely have enough time for this.

7.3.4 Voltage Transformer Sensor

The voltage transformer sensor, in the current iteration of our project, is not something we plan to implement right away. Since our outlet already includes a current transformer sensor, the device will be able to calculate the current being drawn from the outlet. However, this assumes that the voltage across the outlet’s terminals is exactly 120V AC. However, this is not always the case since the true voltage across the outlet’s terminals may vary slightly. In the case that we add this device into our project, we will be able to read the true voltage across the outlet’s terminals and include that value in our calculations. As a result, the power calculations that our project will be capable of doing will be much more accurate and will give the user a much better idea on how much voltage each outlet is drawing at any given point.

One of the main reasons we haven’t decided whether we will include this device or not is the feasibility of fitting this piece onto our PCB. Due to the fact that it is another large component, and that it would ultimately only increase the accuracy slightly, we have decided against including this part unless we have extra time. However, in the case that we have extra time to test and incorporate the part, we will include it. Additionally, adding the component would increase the cost of our project, even though it would only be a few more dollars. Budgeting is something we have to keep in mind when building this device, since we do not have any financial supporters for our project. This means that if any of the components break or are not functioning properly, we will have to directly pay for the cost to replace those parts. For these reasons, we will consider adding this part if time permits.

7.3.5 A more secure application

Another stretch goal we plan to implement would be to implement the safest possible project we can in terms of cybersecurity. This means that we could potentially implement multiple methods to stop potential hackers from accessing the user's information. These methods can range from utilizing a two-step verification system to captcha systems. If we have any additional time to spare at the end of the project, we plan to implement effective methods for cybersecurity, going above and beyond our capabilities to ensure that any potential user of our outlet can feel safe supplying their private information on the application, without the risk of having that personal information stolen.

7.3.6 A bug-free application

Another stretch goal we have is to remove any bugs we possibly can from the application by the end of the project. Obviously, an application without bugs is virtually impossible, especially due to our lack of experience with application design. However, that won't stop our team from trying to create an application with the least bugs possible. Moreover, if we do have any bugs, we want to make sure they do not disrupt the user from using the application as intended. If time allows us, we plan to test various user scenarios and change our code accordingly. With these tests, we can then address each of the situations where the application is not working as intended. Having a product being mostly bug-free is really important, since it ensures that the user will not have any issues with it. After all, the user is the most important person with regards to the project.

7.3.7 Testing the applications easiness

The last stretch goal we have is to have a series of tests with our product, which serve to test how accessible the app is for any user. In case they have difficulty and are not able to operate the outlet in the desired way, we will change the UI and design of the application accordingly. The manual should also help any users with basic questions about how to use the app by indicating how to use the app step by step. However, it would be more ideal if the app is intuitive and results in an easy and straightforward user-experience directly.

7.4 Summary after project completion

When it comes to the administrative part of our project, everything mostly followed the path we had in mind: The budget remained the same as we predicted, since none of the parts had to be replaced, and the timeline we had in mind was followed through. Unfortunately, the stretch goals we had in mind could not be implemented, given the short Summer semester. Even so, we were proud of the work we did during this term.

8.0 Conclusion

This project is one that will benefit several of us. Not only is it something that all of our team members want to work on, but it targets several of our weak points and will give us the opportunity to grow. This is important for us as seniors, since it will help prepare us for a more realistic working world. Additionally, the project has a practical use. Helping people save energy by making them aware of where they are using it is highly practical and has the potential to help many people.

8.1 Project Alternatives

Some of our backup projects include:

- Smart trash can, which is a project with the objective to fully automate a trash can, making it able to move around a set path, finding the most optimal way to the user whenever a button is pressed, or going around a random path, avoiding collisions at all times.

  This trash can would also be able to be connected to an app, where you can choose the type of trash (recyclable or non recyclable) being thrown in, opening the lid for the right compartment and incentivizing recycling for the user.

- Smart Beverage maker, which is a project to fully automate a smoothie machine, by utilizing a conveyor belt and taking the cup from one beverage dispenser to another.

  The user would have a choice to decide which beverage they want via an application, and the machine should automatically decide how much of each liquid it should put on the cup for the correct recipe, leaving to the user only the job of placing and picking up his cup.

- AR Glasses, which is a project to fully build a pair of glasses with a built-in Heads Up Display, or HUD, that would be able to be connected to your phone and show notifications. It would also have transition lenses, meaning the lenses would go darker whenever the user is in a place with higher luminosity and back to normal whenever the surrounding light fades.

  The goal of this project would be to improve on the currently existing smart glasses, by making them more interactive, with a built in voice control system, and a longer battery life.

9.0 Appendices

Below are some additional materials that can help explain where we got some of the materials for this project. This includes both research information that shows how we drew certain conclusions about the project, as well as verification from people about using their materials.

9.1 References

Chandler, Nathan. “How Does a GFCI Outlet Work?” *HowStuffWorks*, HowStuffWorks, 1 Apr. 2000, https://home.howstuffworks.com/question117.htm.

Contact Us, https://www.materialtree.com/bengaluru/contact-us.

“Department of Labor Logo United Statesdepartment of Labor.” *Electrical - Ground-Fault Circuit Interrupters (GFCI) | Occupational Safety and Health Administration*, https://www.osha.gov/electrical/hazards/grounding/gfci#:~:text=The%20ground%2Dfault%20circuit%20interrupter,equipment%20along%20the%20circuit%20conductors.

“Disinfecting Spaces.” *Legrand US*, https://www.legrand.us/.

“ESP32-S3.” *Wi-Fi & Bluetooth MCUs and AIoT Solutions I Espressif Systems*, https://www.espressif.com/en.

“Ground Fault Circuit Interrupters (Gfcis):” *Safe Electricity*, 13 July 2020, https://safeelectricity.org/ground-fault-circuit-interrupters-gfcis/.

Holden, Brad. “Home Wiring Demystified: Electrical Cable Basics You Need to Know.” *Family Handyman*, Family Handyman, 25 Mar. 2021, https://www.familyhandyman.com/list/cable-and-wire-basics/.

“Instrumentation Tutorials: Instrumentation Study Material.” *Inst Tools*, https://instrumentationtools.com/.

KasaSmart.com. “Kasa Smart Wi-Fi Power Outlet, 2-Sockets.” *Kasa Smart*, https://www.kasasmart.com/us/products/smart-plugs/kasa-smart-wifi-power-outlet-kp200.

Megida, Dillion. “What Is JavaScript? A Definition of the JS Programming Language.” *FreeCodeCamp.org*, FreeCodeCamp.org, 28 Apr. 2021, https://www.freecodecamp.org/news/what-is-javascript-definition-of-js/.

Sense. “How Does Sense Detect My Devices?” *Sense Blog*, 30 Mar. 2022, https://blog.sense.com/articles/how-does-sense-detect-my-devices/.

Teja, Ravi. “Classification of Relays: Different Types of Relays.” *Electronics Hub*, 6 Sept. 2021, https://www.electronicshub.org/classification-of-relays.

Thiele, Timothy. “Electrical Code Rules for Every Room in the House.” *The Spruce*, The Spruce, 6 Dec. 2021, https://www.thespruce.com/common-electrical-codes-by-room-1152276.

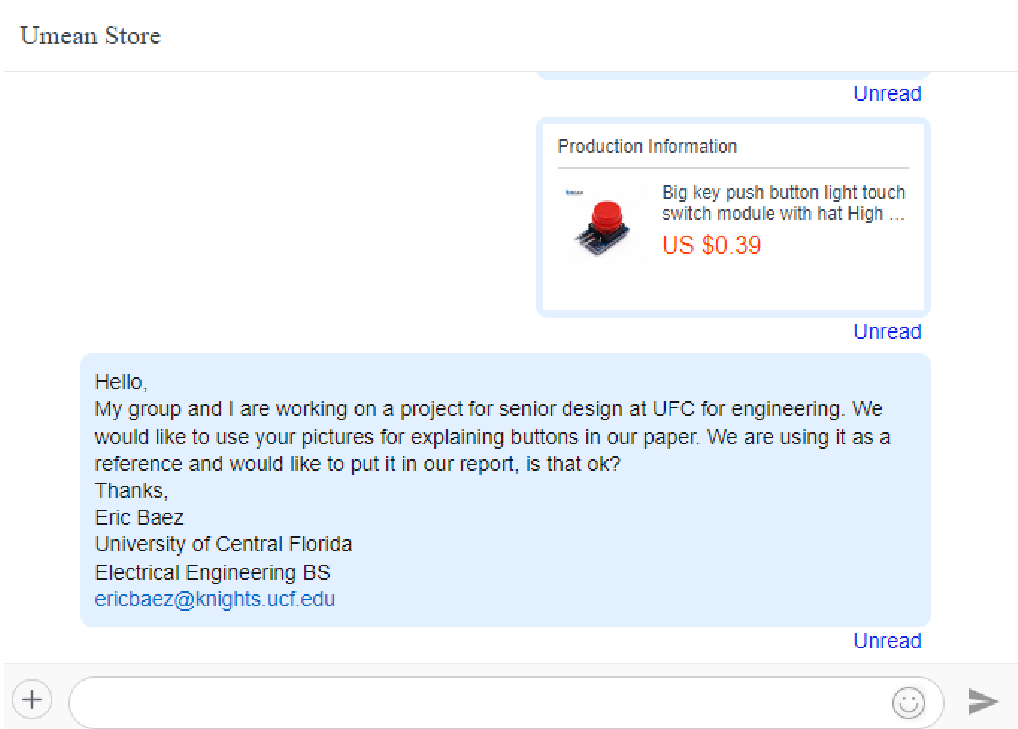
“What Is an AFCI.” *AFCI Safety*, https://www.afcisafety.org/afci/what-is-afci/#:~:text=An%20AFCI%20is%20a%20product,potential%20to%20initiate%20electrical%20fires.

“WIFI Networking Equipment for Home & Business: TP-Link.” *TP*, https://www.tp-link.com/us.

“Electronics Hub.” *Electronics Hub*, https://www.electronicshub.org/.

Upmation. https://upmation.com/wiring-diagrams/.

9.2 Copyright Permission



Eric Baez

ericbaez@knights.ucf.edu

Hello,
My group and I are working on a project for senior design at UFC for engineering. We would like to use your pictures for explanation in our paper. We are using it as a reference and would like to put it in our report, is that ok?
Thanks,
Eric Baez
University of Central Florida
Electrical Engineering BS
ericbaez@knights.ucf.edu

9 + 12 = 21

SUBMIT

YOUR CONVERSATION WITH
Fayleeko Direct

Apr 25, 2022 8:04 PM

Hello,
My group and I are working on a project for senior design at UFC for engineering. We would like to use your pictures for explaining relays in our paper. We are using it as a reference and would like to put it in our report, is that ok?
Thanks,
Eric Baez
University of Central Florida
Electrical Engineering BS
[e-mail address removed]

Apr 25, 2022 8:07 PM

Hello,
My group and I are working on a project for senior design at UFC for engineering. We would like to use your pictures for the energy reading outlet in our paper. We are using it as a reference and would like to put it in our report, is that ok?
Thanks,
Eric Baez
University of Central Florida
Electrical Engineering BS
[e-mail address removed]

Apr 25, 2022 8:44 PM

It's OK, as long as it is not for commercial use.

7. **Eric**

April 26, 2022 at 10:59 am

Your comment is awaiting moderation. This is a preview; your comment will be visible after it has been approved.

Hello, My group and I are working on a project for senior design at UFC for engineering. We would like to use your pictures for explanation in our paper. We are using it as a reference and would like to put it in our report, is that ok? Thanks, Eric Baez University of Central Florida Electrical Engineering BS ericbaez@knights.ucf.edu

Contact Me

Fields marked with an * are required

Name *

Eric Baez

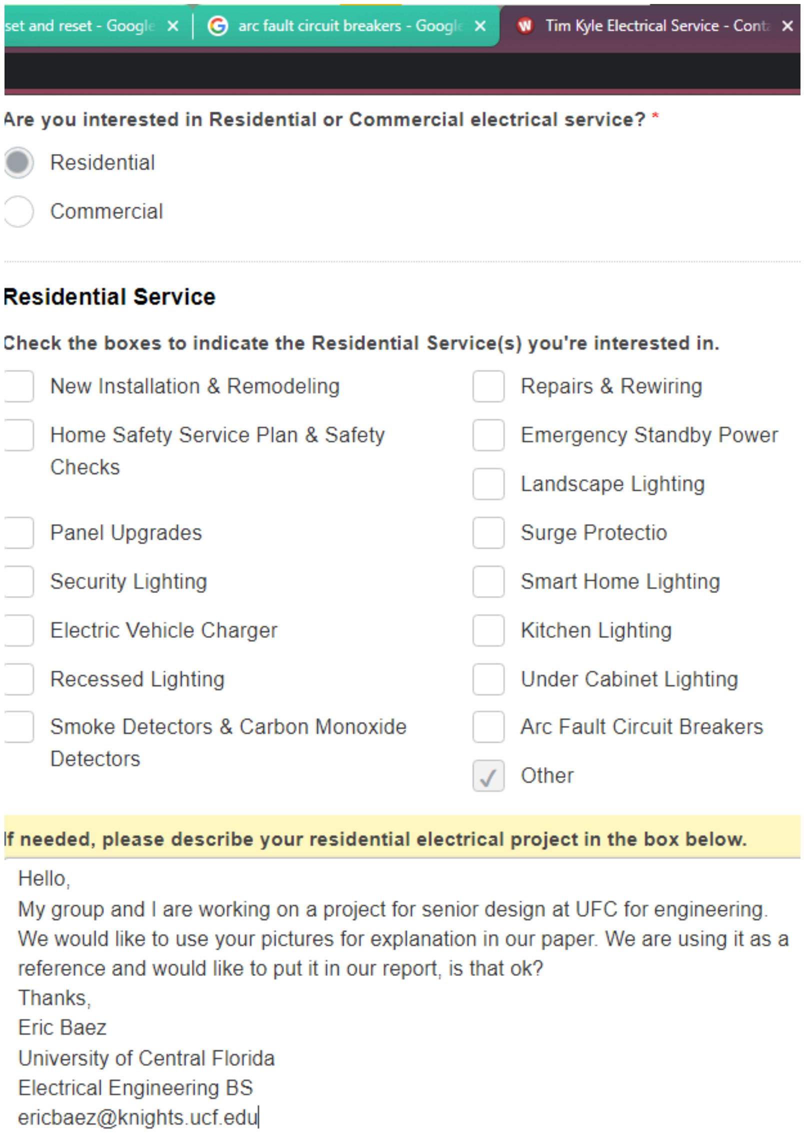
Email *

ericbaez@knights.ucf.edu

Message *

Hello,
My group and I are working on a project for senior design at UFC for engineering. We would like to use your pictures for explanation in our paper. We are using it as a reference and would like to put it in our report, is that ok?
Thanks,
Eric Baez
University of Central Florida
Electrical Engineering BS
ericbaez@knights.ucf.edu

Submit



set and reset - Google ✕ | arc fault circuit breakers - Google ✕ | Tim Kyle Electrical Service - Conta ✕

Are you interested in Residential or Commercial electrical service? *

- (●) Residential
- ( ) Commercial

Residential Service

Check the boxes to indicate the Residential Service(s) you're interested in.

- [ ] New Installation & Remodeling
- [ ] Home Safety Service Plan & Safety Checks
- [ ] Panel Upgrades
- [ ] Security Lighting
- [ ] Electric Vehicle Charger
- [ ] Recessed Lighting
- [ ] Smoke Detectors & Carbon Monoxide Detectors
- [ ] Repairs & Rewiring
- [ ] Emergency Standby Power
- [ ] Landscape Lighting
- [ ] Surge Protectio
- [ ] Smart Home Lighting
- [ ] Kitchen Lighting
- [ ] Under Cabinet Lighting
- [ ] Arc Fault Circuit Breakers
- [x] Other

If needed, please describe your residential electrical project in the box below.

Hello,
My group and I are working on a project for senior design at UFC for engineering. We would like to use your pictures for explanation in our paper. We are using it as a reference and would like to put it in our report, is that ok?
Thanks,
Eric Baez
University of Central Florida
Electrical Engineering BS
ericbaez@knights.ucf.edu

Zulaikha Geer, Research Analyst

Answered Oct 31, 2018

There are three important front end web development languages:

1. HTML
2. CSS
3. JavaScript

Think of HTML (HyperText Markup Language) as the skeleton of the web. It is used for displaying the web.

On the other hand, CSS is like our clothes. We put on fashionable clothes to look better. Similarly, the web is quite stylish as well. It uses CSS which stands for Cascading Style Sheets for styling purpose.

Then there is JavaScript which puts life into a web page. Just like how kids move around using the skateboard, the web also motions with the help of JavaScript.

246 views · View 5 upvotes

5

Add a comment...

Add Comment

Eric Baez · 0y

Hello,

My group and I are working on a project for senior design at UFC for engineering. We would like to use your pictures for explanation in our paper. We are using it as a reference and would like to put it in our report, is that ok?

Thanks,

Eric Baez

University of Central Florida

Electrical Engineering BS

ericbaez@knights.ucf.edu

Upvote Reply

t and | Building Communi | Accounts - GitHub | outlet cover plate | ENERLITES Duplex | Wiring Devices, Lig | How can I under

Name

Eric Baez

Email

ericbaez@knights.ucf.edu

What do you need help with?

Other

Subject

Permission

These articles might be helpful:

- GraphQL API: How to fetch all repositories that the current user has access to
- Permission Denied "exec entrypoint.sh"
- How to create a PAT with write:packages and without repo permissions?

How can we help?

Hello,
My group and I are working on a project for senior design at UFC for engineering. We would like to use your pictures for explanation in our paper. We are using it as a reference and would like to put it in our report, is that ok?
Thanks,
Eric Baez
University of Central Florida
Electrical Engineering BS
ericbaez@knights.ucf.edu

EB Eric Baez
Mon 4/25/2022 11:20 PM
To: enquiry@materialtree.com

Hello,
My group and I are working on a project for senior design at UFC for engineering. We would like to use your pictures for explanation in our paper. We are using it as a reference and would like to put it in our report, is that ok?
Thanks,
Eric Baez
University of Central Florida
Electrical Engineering BS
ericbaez@knights.ucf.edu

InstrumentationTools <instrumentationtools@gmail.com>
Mon 4/25/2022 11:16 PM
To: Eric Baez

Okay, You can use with proper credits.

...

Reply | Forward

Eric Baez
Mon 4/25/2022 10:48 PM
To: InstrumentationTools@gmail.com

Hello,
My group and I are working on a project for senior design at UFC for engineering. We would like to use your pictures for explanation in our paper. We are using it as a reference and would like to put it in our report, is that ok?
Thanks,
Eric Baez
University of Central Florida
Electrical Engineering BS
ericbaez@knights.ucf.edu

Espressif Systems <sales@espressif.com>
Mon 4/25/2022 10:35 PM
To: Eric Baez

Hi Eric,

Thanks for contacting Espressif.

Yes you can use official ESP32 pictures or any public documentation in your paper as a reference. Please just indicating the source.

Thank you!

Best regards,

Annabelle Tian 田柏华
Business Administration | Espressif Systems

Reply | Forward

Eric Baez
Mon 4/25/2022 11:15 PM
To: support@sonicelectronix.com

Hello,
My group and I are working on a project for senior design at UFC for engineering. We would like to use your pictures for explanation in our paper. We are using it as a reference and would like to put it in our report, is that ok?
Thanks,
Eric Baez
University of Central Florida
Electrical Engineering BS
ericbaez@knights.ucf.edu

Reply | Forward

Thomas P Cunningham <tom.cunningham@legrand.com>
Mon 4/25/2022 5:10 PM
To: Eric Baez

Certainly. If you would like anything else please just reach out and we will see what we can get for you.

Best of luck on your project!

Tom Cunningham
Director Product Management, Pass & Seymour
ELECTRICAL WIRING SYSTEMS
13154688097 / Cell 5859199669
tom.cunningham@legrand.com
www.legrand.us

legrand®

Legrand, North & Central America
50 Boyd Avenue
Syracuse, NY, US, 13209

...

Eric Baez
Mon 4/25/2022 8:01 PM
To: support@topgstaging.wpengine.com

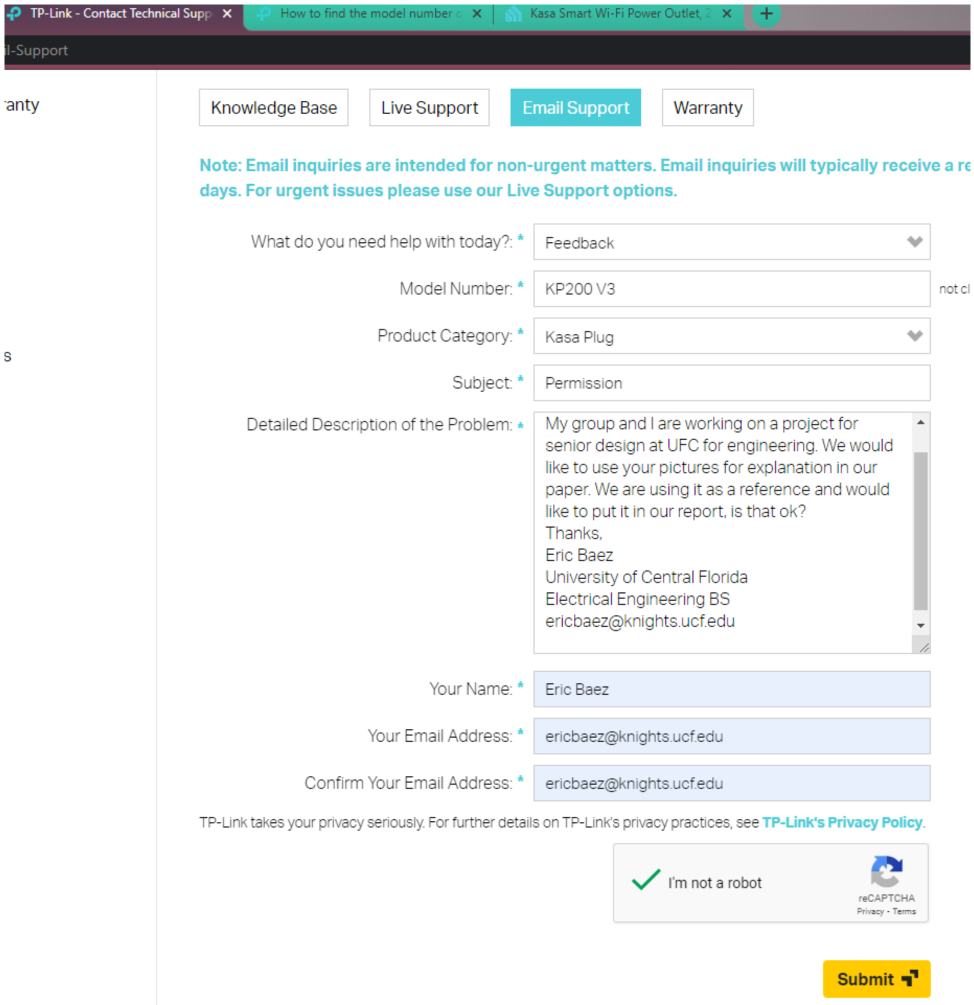
Hello,
My group and I are working on a project for senior design at UFC for engineering. We would like to use your pictures for explaining relays in our paper. We are using it as a reference and would like to put it in our report, is that ok?
Thanks,
Eric Baez
University of Central Florida
Electrical Engineering BS
ericbaez@knights.ucf.edu

Reply | Forward

Eric Baez
Mon 4/25/2022 3:53 PM
To: admin@electronicshub.org

Hello,
My group and I are working on a project for senior design at UFC for engineering. We would like to use your pictures for explaining relays in our paper. We are using it as a reference and would like to put it in our report, is that ok?
Thanks,
Eric Baez
University of Central Florida
Electrical Engineering BS
ericbaez@knights.ucf.edu

Reply | Forward


TP-Link - Contact Technical Supp
How to find the model number
Kasa Smart Wi-Fi Power Outlet,
l-Support
anty
Knowledge Base
Live Support
Email Support
Warranty
Note: Email inquiries are intended for non-urgent matters. Email inquiries will typically receive a re
days. For urgent issues please use our Live Support options.
What do you need help with today?: *
Feedback
Model Number: *
KP200 V3
not cl
Product Category: *
Kasa Plug
Subject: *
Permission
Detailed Description of the Problem: *
My group and I are working on a project for senior design at UFC for engineering. We would like to use your pictures for explanation in our paper. We are using it as a reference and would like to put it in our report, is that ok?
Thanks,
Eric Baez
University of Central Florida
Electrical Engineering BS
ericbaez@knights.ucf.edu
Your Name: *
Eric Baez
Your Email Address: *
ericbaez@knights.ucf.edu
Confirm Your Email Address: *
ericbaez@knights.ucf.edu
TP-Link takes your privacy seriously. For further details on TP-Link's privacy practices, see TP-Link's Privacy Policy.
I'm not a robot
reCAPTCHA
Privacy - Terms
Submit