WiSSP

Wireless Secure Smart Plug

A Major Qualifying Project
Submitted to the Faculty of
Worcester Polytechnic Institute

in partial fulfillment of the requirements for the
Degree of Bachelor of Science
in Electrical and Computer Engineering

Submitted by
Daniel Caffrey
Justin Garon
Joshua Hogan
Dylan Snay
On the 10 of March 2020

Submitted to:

Professor Stephen J. Bitar

Worcester Polytechnic Institute

Table of Contents

Abstract

Smart products have revolutionised the world around us. These devices allow appliances and otherwise ordinary household electronics to communicate with other smart products and the user. This automation can come in many forms, from a refrigerator telling its owner they are out of milk to a thermostat adjusting its level when no one is in the house. This market is still new and has a lot of room for improvement. There are three problems we sought to solve: 1) what if someone wants the convenience of a smart device on a non-smart product, 2) how can a device help the user identify malfunctions, and 3) how can we help assuage one of the greatest fears surrounding smart technology: security.

Our answer to these problems is the Wireless Secure Smart Plug (WiSSP). The product consists of a hub and at least one outlet device. The user interfaces with the system through a smartphone app. The outlet device plugs into the wall and has an outlet that appliances can be plugged into. It consists of a casing with circuitry inside allowing for power measurement and the ability to power on and off the device plugged into it. The outlet device monitors the circuit, sending voltage and current measurements to the hub device, and is able to turn the connected product on or off based on the user's commands. A home could have many outlet devices, but only needs to connect to one hub device. The hub uses the communication protocol Z-Wave to receive and transmit data to and from the outlet devices. This data is then sent to a database, where it's formatted. The user can use the app to look at each outlet's power usage and can turn each outlet on or off, either manually or using a schedule. The hub also uses a predictive algorithm to track the normal power use for each outlet, and if the power being drawn significantly deviates from normal behavior. The app then alerts the user, as this is the most common sign of a device malfunction.

Acknowledgements

We would like to thank all of the WPI faculty and staff for their continued support of all projects like ours.

We would like to thank Professor Stephen Bitar for his guidance and support to overcome problems and seek out optimal solutions.

Additionally, we would like to thank the WPI ECE Shop staff for providing hardware for our design.

The team would also like to thank all those who participated in the market research survey conducted as part of the background research for this project for their patience and detailed input.

Executive Summary

Smart products have begun to change the way people interact with their everyday environment. This market of devices is broad and consists of technology that allows users to directly interact with neighboring devices. One example is a smart refrigerator detecting you are out of milk and sending a reminder to your phone. The market for these devices is growing with no sign of slowing, and as such opportunities for continuous improvement present themselves.

The average home also can be improved by this technology. Sources from the US Department of Energy finds that 75 percent of the power used by appliances occurs when the appliance itself is not in use. An average of $100 a year is lost on such power usage. Devices with functionality such as a standby mode or chargers plugged in without a load also contribute to this wasted power. Malfunctioning sensors or circuitry can cause electronics to run continuously or not run at all. Not all of these faults are easy to spot with the naked eye, but eliminating this waste can save the average homeowner a lot of money.

The ultimate goal of this project was to design a smart outlet system that would provide a homeowner the ability to track the power usage of certain appliances over time, while still providing the normal conveniences of a smart outlet. This includes features such as remote on-and-off control, on-and-off scheduling, and ease of use and setup. This goal was fine-tuned using data from individual users via one-on-one conversation and an online survey that collected 131 responses from various demographics.

Using this data, a list of customer requirements was generated. The greatest concerns among users was security and privacy. Thus, it was important that we designed an outlet that was resistant to unauthorized control and that would prevent third parties from accessing the data collected without permission. Additionally, it was established that the system should monitor and record power usage data, visually display the data to the user, be as easy to set up and use as possible, have the ability to be remotely controlled, and be able to handle large electrical loads for devices such as refrigerators or air-conditioning units. In order to comply with generally expected standards, the system also had to be reasonably inexpensive and comply with FCC regulations regarding wireless transmission.

With these customer requirements in mind, general product specifications were established. The general system would consist of a user interface, via an electronic medium such as a mobile app, and two separate physical devices: a hub and an outlet. This would help address security concerns as well as allow expandability and offline functionality. The user interface would require a database to store the data from the hub, a graphical display to visually represent the data, a user account system to restrict access, a warning when the outlet's power usage deviates from a specified norm, and a scheduler to track and monitor what time the user specifies the outlet should be on and off. The hub would require an ethernet port for internet access, embedded processor for data processing and LAN communication, a LAN interface chip to communicate with the outlet devices, an AC to DC converter for power, a reset button for manual reset, and a status LED to confirm functionality. The outlet would require a small embedded controller to process and transmit data, a power monitoring circuit to provide power data to the controller, an analog to digital converter to convert the voltage and current measurements into a form the controller could process, a LAN interface chip to communicate with the hub, a reset button for manual reset, a pairing button to link it with the system, an AC to DC converter for power, and a relay to turn on and off the power supplied to the connected appliance or device.

The initial design was built around the communication protocol Z-wave. This is a more secure protocol than Bluetooth or conventional Wi-Fi connection. The outlet device was designed to function around the Teensy 4.0 microcontroller. This controller is small and capable of UART and I2C communication to transmit information to the external Z-wave module. Additionally, it contains its own internal ADC so an external one is not required. The circuitry would be supplied from the normal $120V_{RMS}$ line, so an AC to DC converter is required. The design would require a 5V supply, so the converter should have a $5V_{DC}$ output. In order to turn the circuit on or off, the system would use a switching relay and supporting circuitry as shown in Figure E.1. The relay would need to be rated for 20 Amps to ensure that it can handle the maximum current pull from the circuit. To measure the power, two components will be needed: one to measure voltage and the other to measure current. The voltage measurement can be handled by the ADC itself. Since the controller's ADC can only handle a maximum value of

3.3V, a voltage divider was used to scale the voltage by about a factor of 1/60. This means that the controller only needs to scale the value back up via multiplication and the peak voltage can be recorded. The current can be measured using an integrated circuit designed for that purpose. This device would output a voltage that is linearly proportional to the current passing through the device. The controller would then use that ratio to calculate the circuit's current draw. Additionally, the controller will use an external transceiver chip to process and send the data via Z-wave to the hub device.

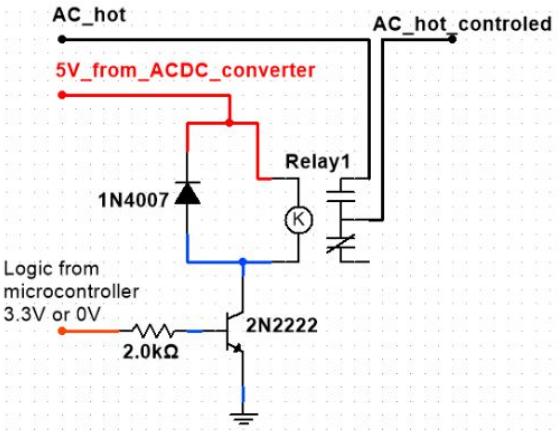

Figure E.1: Relay Control Circuit

The hub device design requires much of the same peripheral circuitry to function. The design would be powered by a $5V_{DC}$ supply that was converted from the standard $120V_{AC}$ from a wall outlet. The controller that is responsible for processing the information received from the outlet was designed to be a Raspberry Pi with an external USB stick that allows the device to function as a Z-wave controller. The device can then monitor the network and retrieve all necessary data sent from the connected outlet devices. Once the hub device has received the data, it is designed to transmit the data via ethernet connection to an online database, where the information can be stored until the user has need for it.

Testing the device went through five phases: hardware testing, software testing, communication testing, microcontroller testing, and system testing. Hardware testing involved characterizing the components, especially the voltage dividers and current sensor, to find how they would affect the signals inputted to them. Using this data, the equations in the outlet microcontroller were adjusted to ensure that the data it was reading was accurate to the circuit. Once the components were working and the system as a whole functioned, the design was used to create a PCB on which the components could be mounted and put into a compact casing for the final design. Software testing involved using the mobile application to visualize and adjust fake generated data to ensure the app behaved as desired. Adjustments to syntax and other problem areas were made as necessary. The wireless communication testing involved reading the

data log from the hub controller and putting controlled valued signals into the outlet controller. The log would display the value transmitted. Microcontroller testing involved taking the outlet controller and inputting a controlled value and ensuring the calculations ran and data transmitted. It was at this testing phase that it was found the external Z-Wave chip was not functioning properly. The chip was determined to need more peripheral circuitry and computing power than was available. Thus, a different solution to transmit the data to the hub was necessary. A replacement controller for the Teensy was found: the Z-Uno. This Arduino-based controller has built-in Z-Wave transmission capabilities, removing the need for an external functioning chip. All of the peripheral circuitry is the same as the Teensy, so no other adjustments to the design needed to be made other than pin assignments and firmware code. Finally, the system as a whole was connected and tested. The outlet and hub devices were plugged in and connected. Then a known load was plugged into the outlet and the power usage was measured and recorded using the mobile application.

For future improvements, it is recommended that serious noise reduction measures are implemented to improve the accuracy of the measurement devices. Isolating the AC and DC neutral and ground would improve device safety. A latching relay instead of the normally connected relay would decrease power loss in the design. The microcontroller and ADCs for the outlet are set up using internal references. Having a regulated external voltage reference or an ADC with improved accuracy will produce more accurate overall results. Implementing more features into the Z-wave management software would improve the user experience and further improve design simplicity. Finally, the mobile application is designed for android and its security is limited by such. Adding more application security measures will better increase the security of the overall system.

In conclusion, the product that was built functions as desired. The outlets can be individually monitored and turned on or off, either manually or via a schedule. While the product is functional to a degree that already improves on the existing market, numerous improvements would still need to be made to bring it to market. Once these are implemented, however, this product will be a secure and unique addition to the smart-home industry.

1. Background

1.1. Definition of a Smart Device

Smart products have begun to revolutionize the way people interact with their environment. A “Smart Device”, also interchangeable with “Intelligent Device,” is defined as “an electronic gadget that can connect, share and interact with its user and other devices” (Technopedia, 1). These devices can adopt behaviors provided by the user or follow the commands of other smart devices. This behavior can be explained with three major categories. For a product to be considered a smart product it must be able to adapt to (Maass, 3):

- Various Situational Contexts
- Users who interact with the devices
- Manufacturer needs and business requirements

Smart products are meant to adapt to the physical environment, but are also intended to adapt to the local smart home environment. This appeals to the population due to its customizability and variability depending on each smart home or user, as well as its convenience. To go even further, the three points above can also be divided into six other points. Following these guidelines a smart device must be (Maass, 3):

- Situated - Can recognize situational and community contexts
- Personalized - Can change features and settings based on the user’s needs.
- Adaptive - Is able to change based on buyer and consumer reviews.
- Proactive - Able to anticipate a user’s habits and intentions.
- Business Aware - Follows and considers business constraints.
- Network Capable - Can interact and connect to other devices.

These come together to create the backbone of a smart device: A device that can adapt to its environment and user, as well as connect to multiple other devices for easy use and customization. Every smart product includes these specifications. For example: a smart outlet can be programmed to the user’s intent, interact with other devices, and can be remotely updated.

Any device can be smart as long as these guidelines are all met. This type of design has created a way for homeowners to completely customize their technical living style, as well as decrease the time users spend on mundane tasks.

Smart home products often center around a service that links individual devices together. The main two systems on the market are the Google Nest and Amazon Alexa Systems. Each employs a service to connect and control all the smart devices in your home either with an app or using voice commands to certain smart devices, usually a smart speaker (Gebhart 2019) . Each service is compatible with thousands of brands of devices, and allow you to automate functionality of those devices through routines of scheduled actions. Smart devices range from light bulbs to thermostats to appliances to wall plugs. For example, according to their website, Google Nest employs smart technologies in Home Assistance, Security, Entertainment, Energy, Connectivity, and Lighting. Devices in these fields all connect over the Google Nest Service, and can be controlled and automated by either their phone app or by voice control with one of their smart speakers.

1.2. Current Market Analysis

Market research has shown that smart products are not just extraneous home appliances, but are desired by millions of consumers. In the year 2019, the US revenue of smart products was around 28 Billion dollars (Statistica, 1). It is profitable to enter the smart product market. In 2019, each home equipped with smart tech contributed an average of $113 to the market. There are currently 43 million smart homes worldwide. The market is at an all time high, yet is still projected to grow by a rate of around 16.9% from 2019 to 2023 (IDC, 1). By 2023 it is predicted that the market will reach a volume of around 44 billion dollars, almost double to what it is in 2019.

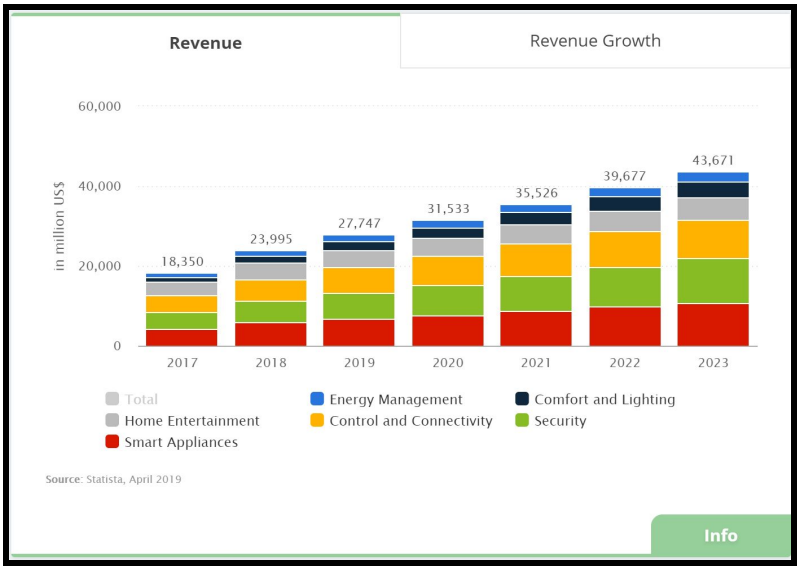

Figure 1.1: Smart Market Revenue Predictions (Statistica, 1)

While the market for smart home products is still new, there are numerous smart outlet systems that have already been created to fill the void. In order to develop a system that we can market and that is the best product possible, we conducted research on some of the devices that are already on the market.

Some competing systems on the market are integrated systems, also called Protocol-Specific systems. These devices are made to be put in place of normal wall outlets or into newly constructed buildings. They require experienced electricians to install and work with the building's wiring and electrical infrastructure. Products like this are comprehensive, measuring vast amounts of data to both users and the company servers they use. The downside is that these systems are expensive and are far from convenient to install. They are designed to be implemented into newly constructed buildings or buildings that are undergoing massive renovation. As such, it's audience is not our target (Postscapes, 1).

The systems we are planning to model our product after are the plug-in systems. Plug-in system smart plugs are often one of the first smart devices consumers will purchase as they are versatile (Brown 2019). They can be used on anything that plugs into a wall, and can be used to turn a normal device into a smart device, meaning a consumer doesn't have to buy a device with smart features built in. Nearly every smart plug focuses on two main features: First, the ability to

turn power to a device on or off using either voice commands or an app, and second, the ability to set routines so that power to a device can automatically be turned on or off at certain times (O'Boyle 2019). Some plugs, such as the Currant Smart Plug, also focus on other features such as power monitoring. As previously mentioned, these devices can also be connected to a system hub, such as the Google Nest.

The specific smart outlets evaluated were the Bayit Wifi Socket, Belkin WeMo Insight Switch, Conico Mini WiFi Smart Outlet, Eve Energy, iDevices Switch, iHome Wifi Smart Plug, Orvibo Wifi Smart Socket, and TP-Link Smart Plug. All these devices are sold on Amazon and have a price range of $7 between $80. Upon evaluation of the user complaints and comments with these systems, there are a few common issues that arise. The first is that the users complain because the features are not adequately explained or unclear. The main things a user expects of systems are that they can be remotely controlled by some means, can be automatically scheduled by some means, and can monitor the amount of power being used. The ideal breakdown of the power data could not be gathered from the reviews. The second main complaint is that the system needs to be reliable. Many of the lower-priced systems do not connect to the hub system properly or give errors when users try to connect them. Finally, some users complain that they had to buy a separate hub to use the outlets. Some of the outlets require the Amazon Echo or Google Nest to connect to, causing an extra expense to them. Some of the complainants did not know they needed an extra system to begin with (Amazon Reviews).

This leaves our development with a clear goal. We want to ensure that our device has the basic functions that customers want. Any additional functionality is even better. The system should also be reliably connected and, ideally, be a stand-alone system.

1.3. Energy Consumption and Waste

Smart outlets have the potential to dramatically reduce energy usage if implemented on the correct devices. Homes all across the country waste energy in massive amounts without even knowing it. According to the Berkeley Lab, the typical American home has “forty products constantly drawing power” which amounts to almost ten percent of total residential electricity use (Meier Data, 1) and equates to over $100 per year (Dew, 1). The US Department of Energy claims that seventy-five percent of the energy used by appliances is used when they are turned off (Miller, 1). This phenomenon is known as standby power.

Standby power happens because many devices that are switched “off” or appear to be off are still drawing power for background functionality, both desired and not desired. Common offenders that consume a large amount of standby power are TVs and accessories, computers, and chargers without load. This standby power can be eliminated by disconnecting the devices from their power supply when not in use; however, it is difficult for homeowners to know which appliances are the worst offenders (Meyers, 568-9).

Additionally, energy is often wasted when devices are not operating properly or are left on by accident and go undetected for large periods of time. Many devices and appliances have sensors to determine when they should be on and when to be off. Sometimes these sensors will not function properly. For example, a man had an ice melt system installed on his roof to prevent ice dams from forming. Unbeknownst to him, the sensor that regulates the system and turns on the heat when temperatures fall below freezing was not functioning properly. This caused the system to be running non-stop for several years (Bitar). Even when devices are functioning properly, forgetfulness by the residents of the home can cause waste. For instance, someone may leave a window open when a heater is running in the winter causing the room to never reach the set temperature and the heater to run non-stop. Monitoring the power consumption of devices and systems like these can allow the homeowner to be aware of when they are behaving differently than normal operation. Situations like these cost homeowners extra on their electricity bill and are a detriment to the environment.

1.4 Communication Protocols

Wired devices have various protocols available for transmitting data along a wired connection. These protocols, called Serial Interfaces, are used to transmit large data buses to the memory of a microcontroller or other devices. There are many options, but the options that the Z-wave module uses are the Inter-Integrated Circuit (I2C) and Universal Asynchronous Receiver/Transmitter (UART) protocols. Using these two protocols, the microcontrollers of the Hub and the individual outlets can send and receive data to each of the Z-wave modules, allowing for seamless wireless communication between the Hub and outlets.

The first of the two protocols is the I2C. This protocol requires two pin connections, one forming a clock connection, designated the SCL pin, and the pin for sending the data, designated the SDA pin (Frenzel, 71). An example of the configuration is shown in Figure 1.2.

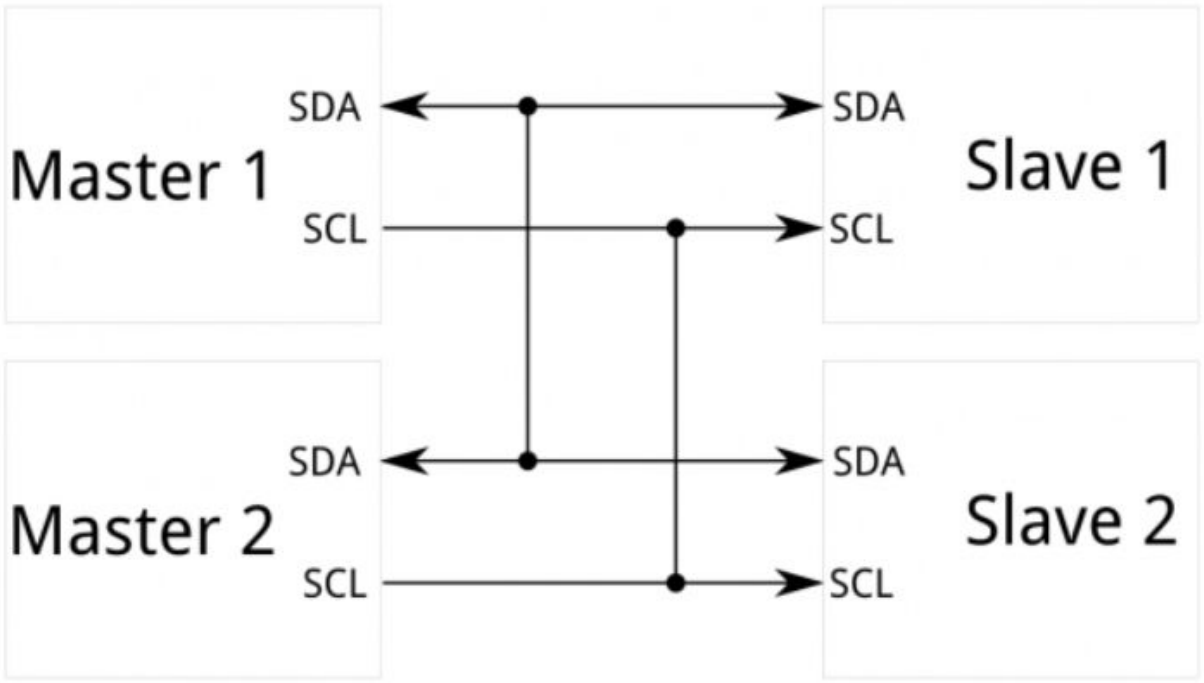

Figure 1.2: I2C Device Configuration with Two Master and Two Slave Devices

In the case of the smart home design, the Master device is the microcontroller, be it the Teensy or the Raspberry Pi, and the Slave device is the Z-wave module. The Master device decides the clock signal for the system. This clock controls the sending pattern for the actual data bus. When the start bit is received into the system, the Master transmits 7 or 10 bits, one bit at a time with the SCL, that correspond to the location of the data in the Slave device's memory. That is followed by an indicator bit, which selects whether the memory location referenced is being written to or read by the Master device. The system then waits for an ACK/NACK signal from the Slave device. After that signal is received, the appropriate device, either the slave if the Master is reading data or the Master if the Master is writing data, begins transmitting the data on the SDA line. This transmission is synchronised with the SCL clock. After eight bits of data is transmitted, there is another ACK/NACK request. This is repeated for the required number of data bytes until a stop bit is received (Frenzel, 71).

The second of the SI's is UART. This is not exclusively a communication protocol, but must be a physical circuit within your two devices. This is a very basic communication scheme that requires only two connection pins between the devices, a TX and RX pin. The full configuration is shown in Figure 1.3 (Circuit Basics, 1).

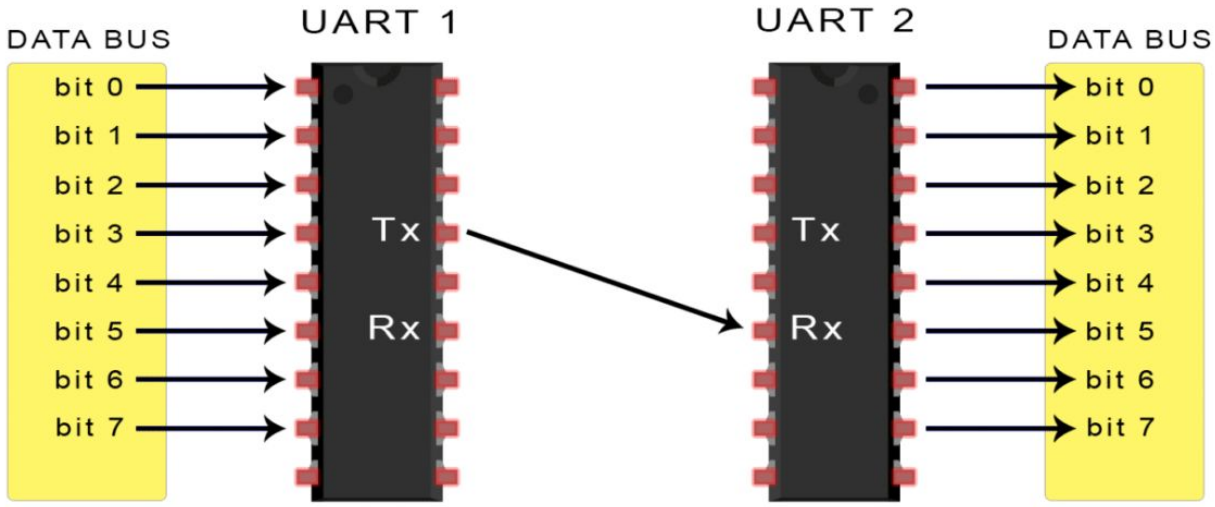

Figure 1.3: UART Communication Setup

When the first UART wants to transmit, they begin sending the data via the TX pin. The data must be sent one bit at a time. However, there is no way to synchronise the sending as they have no clock connection. Thus, the two circuits must agree on the rate at which the data is sent

and the length of the data message. That is the biggest downside to this communication method. Once the data is sent, the UART outputs the reading to all of its individual pins (Circuit Basics, 2).

1.5 Standards

Standards for electrical devices have been developed by numerous government, commercial, and scientific bodies. These exist for various reasons, including user convenience, commercial fairness, and safety for both user and manufacturer. As we develop this project, our design must take into account these many standards not only to qualify for product status on US markets but to be convenient and safe for our product's users. This may change the initial design based on what was initially planned.

1.5.1 Frequency Allocation

In the United States, there are two different governmental bodies that allocate wireless frequency transmission: the Federal Communications Commission (FCC) and the National Telecommunications and Information Administration (NTIA). These two bodies have jurisdiction over which frequencies on the wireless spectrum are used for what purpose. This information is published in the "FCC Online Table of Frequency Allocations". They also have ultimate control over what organizations and groups qualify to use the reserved frequencies. To date, the frequencies from 9kHz to 275GHz are allocated. In our case, we need to ensure that our wireless communications, utilized between the hub and the outlets, do not interfere with any of the reserved frequencies on the spectrum. Such interference, either intentional or unintentional, is illegal (FCC et. Al, 1).

The wireless transmissions our devices will utilize is Z-Wave. As restricted by the FCC, these transmitters operate at 908.4MHz or 916MHz (Silicon Labs). In the above mentioned FCC table, these frequencies fall within a range dedicated to radiolocation and amateur radio use. Since these standards are maintained by the manufacturer and not our team, this will not interfere with restricted frequencies (FCC et. Al, 1).

1.5.2 Electrical and Electronic Equipment Compliance Requirements

Sections in the United States code are designed to help ensure that both manufacturers and users of products are safe. In the case of electronic devices, the Department of Commerce has created a guide specifically to navigate the very many laws that apply. Most of these laws are written to govern certain appliances, but there are a few general laws that all electrical devices must conform to. Firstly, the Customer Product Safety Commission (CPSC) is responsible for ensuring that any device doesn't contain hazardous materials and is not generally harmful to its users. Were we to bring this product into mass production and market, the CPSC would inspect the product and issue a certificate ensuring that each item produced met their safety standards. Additionally, the electrical data must be printed on a label and stuck to the product. This data includes the operational voltage and currents, power efficiency, average energy use, and other elements as prescribed by the Federal Trade Commission (FTC). These values are required to be within certain ranges, as governed by the Energy Conservation Program for Consumer Products. The specific values vary based on what the product is classified as, which cannot be determined without the help of specific Department of Energy (DOE) officials (Benson et. Al, 2).

Additionally, while not federally regulated, many states have a Restriction of Hazardous Substances directive (RoHS). The specific substances vary state to state, but one of the most common restrictions is lead. Any product developed to be sold in the states with this RoHS must have a concentration of lead lower than 0.1%. While most domestic electronic manufacturing utilizes little to no lead, we must ensure that any non-domestic materials we order also comply with these standards. Thankfully, the most common place to find lead is in lead-tin solder, used in making very cheap or old electronic equipment. Provided we purchase parts that are of decent quality, this will be no issue (Benson et. Al, 30).

2. Design Requirements

There were many factors to consider when determining the design of the smart outlet device.the smart outlet device. A survey was conducted to understand the market and to determine what features are most important to consumers. Using this information, product requirements and specifications were determined.

2.1. Customer Requirements

Through a survey, research on existing products, and customer reviews, customer requirements were determined. The survey questions can be found in Appendix D. It shows that a major concern among all respondents was security. Additionally, there was a significant interest in monitoring the power of devices. The following explicit customer requirements were established:

- Monitor Power
- Detect Issues
- Connect with Hub and Application
- Make it Cuztomizable and possibly programmable
- Handle large enough loads
    - Fridge
    - AC
    - Etc
- Remote Connection
- On/Off Scheduling
- Visuals of the Outlet Power Usage
- Follow Behavior of Outlet

Through market research and comparison to other smart outlet products, the following implicit customer requirements were established:

- Inexpensive
- Simple Setup
- Secure

- FCC and NEC Compliant
- Looks good
- Continue to collect power data if internet disconnects
- Restarts when if power goes out

2.2. Product Requirements

From the customer requirements shown in the previous section, the following products requirements were determined.

Power Monitoring - The device should measure the power used at the outlet to be sent to the hub. Our target customers are concerned with being able to see how much power their home appliances are consuming.

Simple Setup - The device should be extremely simple to set up. One of the features ranked highest in the survey was easy setup. Not all of the customers will be technologically inclined. It is important to them that the device can be set up on their own.

Remote Connection/Control - The user should be able to control the behavior of the different outlets remotely. One of the main features of a smart outlet is the ability to control it remotely. One of the main features of a smart outlet is the ability to control it remotely.

Low Power Usage - Naturally, a device targeted at consumers who care about the amount of power being used in their home should not use a lot of power itself. The power consumption of the outlet should be minimal.

Competitive Pricing - This device should be priced such that it is comparable to other smart outlet systems in the market. The target market is the average homeowner,so we shouldn't expect that someone would break the bank for this system. The design should be planned as such.

Track"normal" operation and flag obscurities - The device should be able to judge the normal operation of the device plugged into it and alert the user if the device begins to deviate from this norm. This way a user can tell if there is a problem with the device that would cause excess power use.

Security - The device should be secure, not allowing the users personal data to be stolen or accessed by anyone but the user. This was a large concern among the survey respondents, making it a priority in the design of the system.

Local Storage of data - The product will have some storage of data on the hub in the case that the internet connection disconnects.

Online storage of long term data - store long term data on the cloud so that the user can determine trends, etc.

Minimal Size - Be able to fit behind household appliances.

Local In-Home Connection - The hub and the outlets will be connected through a local area network that is secure and cannot be accessed by devices not in our system.

2.3. Product Specifications

Upon taking into consideration the product requirements, it was decided that the best way to implement the system was to divide into two devices. The system will consist of multiple outlets that connect locally to a singular central hub. This helps to satisfy the requirements of network security, and offline capabilities. The outlets and the hub will communicate over a secure local area network. The two devices will have their own specifications.

2.3.1. Hub specifications

The following are the product specifications for the Hub:

Ethernet Port for Internet Connectivity - An ethernet connection will connect the hub to the web. This will be a secure and easy way to set up the system.

Embedded Processor - An embedded processor allows for low-power processing of outlet schedules and commands from the user app as well as power calculation and both LAN and internet communication.

LAN interface chip - A Tx/Rx chip is needed to implement a LAN protocol such as Bluetooth, or ZigBee in order to achieve the low power, secure in-home network.

AC/DC Converter - The outlet AC power signal must be converted to DC for use in the DC components of the device.

Reset Button - A reset button is a mechanical failsafe that will restart the device to its factory settings. This increases security as the user will be able to restart the system in the case of an unwelcome person having access to the system.

Pairing Button - A button that can be pressed at the same time as an outlet button so that pairing the outlets is easy and quick.

Status LED - A status LED helps the user to confirm that the system is running correctly.

2.3.2. Outlet Specifications

The following are the product specifications for the outlet.

Embedded Controller - a microcontroller to handle the transmission of power data. Size is an issue since the outlet device needs to be as small as possible for convenience.

Power Monitoring Circuit - registers the power being used at the outlet level and provides the data to the microcontroller for transmission

Analog to Digital Converter (ADC) - An analog to digital converter is needed to convert the analog measured voltage and current to digital input for the microcontroller to process/transmit.

LAN Interface Chip - interface with the microcontroller and transmit the required data to the Hub

Reset Button - A reset button is a mechanical failsafe that will restart the device to its factory settings. This increases security as the user will be able to restart the system in the case of an unwelcome person having access to the system.

Pairing Button - A button that can be pressed at the same time as the hub button so that pairing the outlets is easy and quick.

AC/DC Converter - The outlet AC power signal must be converted to DC for use in the DC components of the device.

Relay - A relay is necessary to allow the microcontroller to turn on and off the AC power supplied to the load.

2.3.3. User Interface Specifications

The following are the product specifications for the user interface.

Database - A database is required so the microcontroller can store collected data into a remote location for the app to access. The system also requires a way for the app to communicate with the microcontroller, which is done through the database.

Graphical Display: Required for the user to understand and visualize their power usage in terms of power and cost over time allotted.

User Support: An account system is necessary for the user to create a system with access to the database, also used to support security.

Power Warning Alarm: Alerts the account owner when an outlet deviates from past behavior.

PIN Codes: Allows the user to add extra security to their outlets or hubs to prevent use even outside of the owner's phone.

Scheduling: App will have scheduling capabilities for each outlet to provide greater smart house and component customization.

3. Preliminary Design Approach

Our initial design for the WiSSP smart outlet system network consisted of a centralized hub that communicated with each device via Z-wave. The user controlled the system through a mobile application which updated a database that was accessible by the hub. The hub controls and receives power information from each outlet. Instead of connecting to the home's wifi network, which is more vulnerable to cyber attacks, the Z-wave network between the outlet and hub is isolated. The only connection to the internet is through ethernet. Z-wave was chosen due to its low power and security benefits. For more details on this decision, see Appendix E. A visual representation of the mesh network structure can be seen in Figure 3.1.

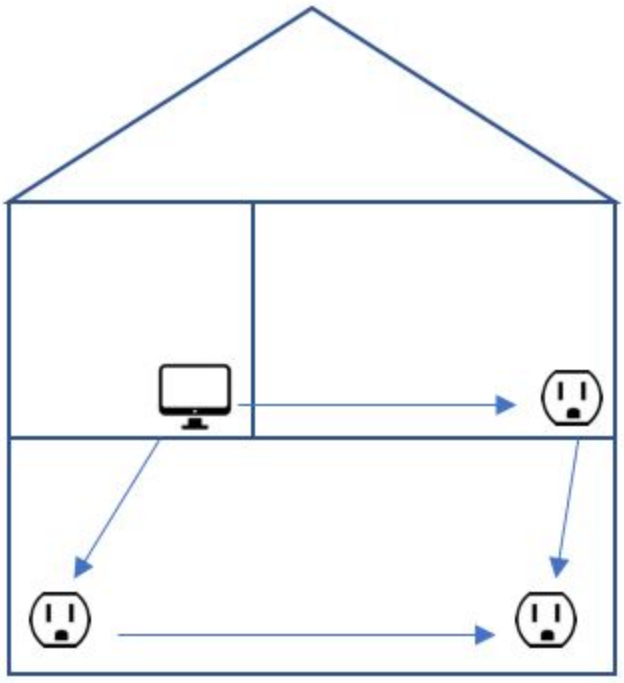

Figure 3.1. Z-Wave Mesh Network

Figure 3.2 shows a block diagram of each submodule of the system. Each submodule will be described in the following sections. Each block was designed to meet the product specifications.

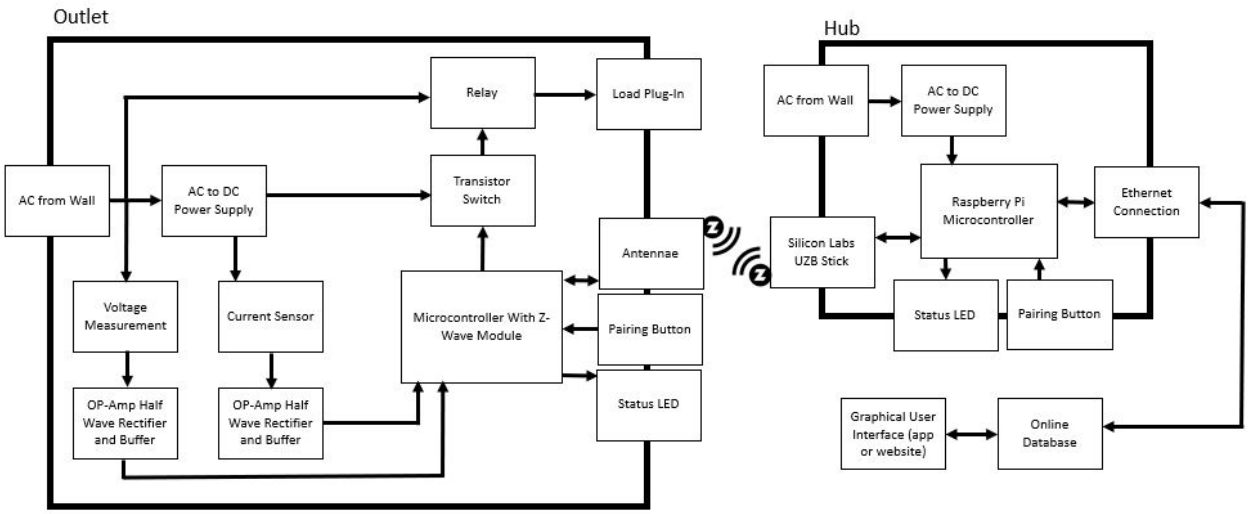

Figure 3.2: System Block Diagram

3.1. Outlet

The outlet device consists of blocks of circuitry that each serve a specific purpose to satisfy the product specifications of the system. The device is designed around a Teensy 4.0 microcontroller that communicates with the subsystems of the outlet and with the hub. The smart smart outlet is powered directly from the AC power of the wall outlet it is plugged into. Together, the blocks work together to monitor the power being used by the load, and control whether it is on or off based on user input.

3.1.1. AC/DC Conversion

The sub-circuitry of the outlet is powered by $5V_{DC}$, but the power for the outlet is being supplied from the $120V_{RMS}$ line. In order to utilize this AC power to supply the sub-circuit, an AC/DC conversion circuit is needed. The design uses a predesigned chip that converts $120V_{RMS}$ at 60Hz to $5.0V_{DC}$ and supplies at least 500mA to the sub-circuit. The decision to use a chip instead of designing a conversion circuit from scratch is with safety as the number one concern. A pre-made chip minimizes the potential for human errors such as soldering errors, which can cause short circuits and fires.

3.1.2. On/Off Control

The outlet is able to switch the load on and off through the user interface. In order to perform this switching, a relay is used. An electromagnetic relay is utilized over solid state due to its low cost. It is not necessary to use a solid state relay since the required switching speed is not high. The T9AS5D22-5 is used to control the $120V_{RMS}$ since it is rated for 20A and standard wall outlets are rated for 15A. In order to power and control the relay coil, it is designed in a configuration with a resistor, a switching BJT, and a flywheel diode as seen in Figure 3.3.

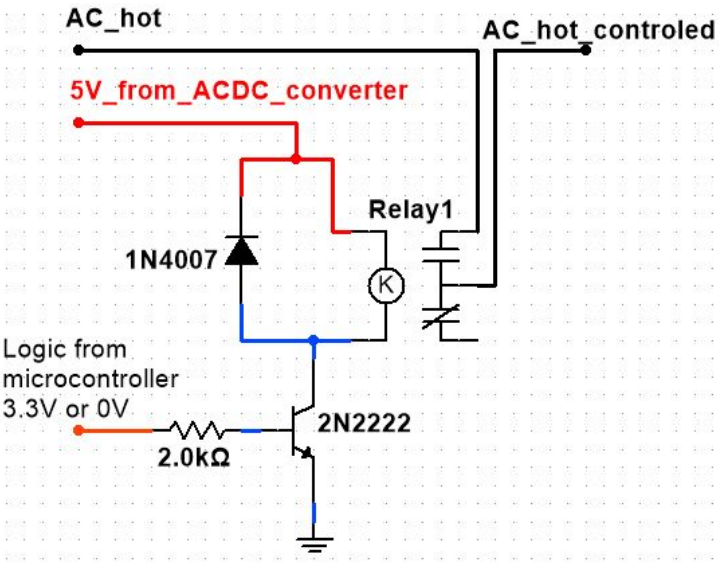

Figure 3.3: Relay Control Circuit

The BJT is necessary so that the current drawn by the relay is coming from the AC/DC converter and not through the microcontroller, which cannot supply the required amount of current. Additionally, the output of the microcontroller is 3.3V and the relay coil requires 5V. The BJT makes it possible to control the relay this way. The flywheel diode is necessary so that the magnetic field on the coil can dissipate when not being supplied. The $120V_{RMS}$ line is connected to the switching terminals of the relay. The control to the relay is connected to the microcontroller through the BJT so that a logic 1 or 0 from the microcontroller will switch off and on the $120V_{RMS}$ line. Since this product is designed for homeowners to monitor the power of their appliances, it is reasonable that the appliances will be on most of the time. For this reason,

the relay will be normally connected so that it doesn't draw power while the appliance is operating.

3.1.3. Power Measurement

The primary purpose of this device is to measure and monitor the power used by the load plugged into the outlet. This will be implemented by measuring the current and the voltage independently and calculating the power from those values. The voltage will be measured using a high resistance voltage divider from the $120V_{RMS}$ line and the neutral line (circuit ground). The voltage divider will be about a 1/50 factor so that the measured voltage will be below 3.3V, as the chosen microcontroller can't receive an input over 3.3V. In order to not load down the voltage divider with the current drawn by the input, an OP-Amp configured as a unity gain buffer. will be used between the voltage divider and the input to the microcontroller. The op-amp will be configured to act as a half wave rectifier (with the negative supply rail at ground) since the microcontroller can only receive a positive input. The microcontroller has built in analog to digital conversion capabilities to measure the peak of the voltage signal.

The current will be measured using the ACS723 current monitoring integrated circuit. This component is a sensor that can measure up to ±20A and give a linear output of 100mV/A. The output will not exceed 3.3V, the maximum of the microcontroller input. The output of the sensor will be fed through an OP-Amp configured to be a one-to-one buffer and half wave rectifier so that the input to the microcontroller is between 0 and 3.3 volts. Since the output of the ACS723 is a function of the supply voltage of the chip, the 5V supply voltage needs to be measured by the microcontroller. Since the input pins of the microcontroller can only handle up to 3.3V, a voltage divider is used to cut the voltage in half to be measured and used for calculation of current.

3.1.4. Physical Interface

The outlet will have features that allow the user to interact with the outlet directly. There will be two buttons: one button will be used for pairing with the hub, and another discrete button to reset the device to factory settings. To connect to the hub, both the pairing button on the outlet

and the button on the hub will need to be held at the same time until pairing is complete. The outlet will also have a status LED that shows the user the status of the device (powered, paired to hub, etc.).

3.1.5. Main Controller

For the outlet, the priority traits in the microcontroller are size and I/O functionality. Due to the microcontroller being the largest component in the casing, it must remain relatively small in order to keep the outlet apparatus as unobtrusive as possible. It only needs to have a small number of I/O input pins, with a few optional analog pins for the power data. It is also necessary for the board to have 5V and 3.3V supply pins. This allows for a greater range of peripheral parts for the sensors and wireless modules.

Upon meeting these requirements, the next important consideration is the memory space. Often small microcontrollers have a small amount of memory, so programming the firmware to fit the storage needs will be taken into account ahead of time. The way the system is structured, the data processing won't utilize the outlet processor's resources. Thus, the responsibilities for the outlet controller are limited to the transmission of power data and the state of the control. None of the data storage or processing should take place on the outlet. With that in mind, Figure 3.4 shows the basic structure for the controller's pseudocode.

```
Pseudocode for Outlet Microcontroller

#include necessary libraries

initialize the correct TX, RX, Program, Sensor Data, and Reset pins
initialize any other required variables

void setup(){
        //sets up necessary setup functions
}

void loop(){
        //reads the Current and Voltage data from the sensor
                //packages the data for transmission
                //sends the data to the hub
                //waits X amount of time before rerunning the program

        //receive transmission from the hub
                //read the binary value transmitted to determine if the outlet is on or off
                //wait for next transmission and recheck the state
}
```

Figure 3.4: Outlet Microcontroller Pseudocode

The other major consideration is cost. In a production environment, the outlet is the unit that will be reproduced the most. In order to make this method of production cost-effective, it should also be the cheapest microcontroller. The microcontroller used in the design, the Teensy 4.0, costs $20 per unit. This could be reduced by bulk purchasing. This is much less expensive than a normal Raspberry Pi model, which costs around $35 per unit.

3.1.6. Wireless Interface

The Z-Wave module located in the outlet consists of a transceiver, which sends and receives Z-wave communication, and a minimal amount of digital circuitry to communicate with the microcontroller. This module is used to create the secure local connection between the hub and outlet. The module used is the ZGM130S Radio Board from Silicon Labs. This module incorporates the ZGM130S transceiver chip with a radio board which contains a coaxial port to connect to an antenna, RF matching circuitry, supporting digital circuitry, and pin connectors to allow for easy access to any pin on the chip. The board comes fully integrated and ready to connect. It connects to the microcontroller through a UART, and is powered via a 3.3V input from the microcontroller. It operates in a lower power mode when waiting for a transmission, and a higher power mode when sending/receiving information.

3.2 Hub

The hub functions as the central unit of the device. It connects to the internet via ethernet connection, allowing it to interface with the internet database. From there, it sends any commands and receives information to and from the outlets via a Z-Wave connection. The hub consists of four physical blocks.

3.2.1. AC/DC Conversion

Similar to the outlet, the sub-circuitry of the hub is designed to be powered by $5V_{DC}$, but the power being supplied is $120V_{RMS}$ from the wall. In order to utilize this AC power, the same AC/DC conversion method is used. The design uses a predesigned chip that converts 120Vrms at 60Hz to $5.0V_{DC}$ and supplies enough current for the subcircuit (at least 500mA). The decision to use a chip instead of designing a conversion circuit by scratch was made with safety as the number one concern.

3.2.2. Physical Interface

The hub has physical interfaces similar to the outlet for the user to interact with. There will be a pairing button that the user will press at the same time as an outlet button to pair the two together. There will be a reset button on the hub to reset to factory settings. There will be an LED to show the status of the hub.

3.2.3. Main Controller

The controller used in the hub is more complex than the ones used in the outlets. The hub microcontroller handles the data processing for the entire system of plugs. It also connects to the internet via ethernet to allow data transmission. In essence, this microcontroller is much more focused toward memory space and clock speed than the more basic microcontrollers used in the outlets.

Figure 3.5 shows pseudocode for the hub system. This program is more intricate than the outlet program, mainly because this needs to handle the scheduling for all of the outlets. The current time is pulled from online, and the hub will cross reference the schedule with the current time to determine which outlets should be on or off. The program also handles the data received from the individual outlets and packages it with a timestamp to be uploaded to the database.

```
Pseudocode for Hub Microcontroller
#include necessary libraries

initialize TX, RX, Program, Reset, and Internet pins
initialize all other variables

void setup(){
        //run any necessary setup functions
}

void loop(){
        //check time
                //this will be done via the internet
                //if internet connection is lost, a temporary counter will be implemented

        //adjust outlet on-off binaries
                //Change the settings based on the schedule that is already set by the user
                //Transmit the binary signals to the appropriate outlets
                //recheck at set interval

        //Upload the data from the outlets
                //Receive data transmissions from each of the outlets
                //Map the data to the specific timestamp
                //Transmit data to the internet database
}
```

Figure 3.5: Hub Microcontroller Pseudocode

3.2.4. Wireless Interface

The Z-Wave module located in the hub will consist of a USB Z-Wave controller stick, which is used to create and manage the Z-wave network. The Silicon Labs UZB Stick is used. The Z-Wave controller is powered by and communicates with the Raspberry Pi through a USB port. It is initialized and can be operated via the command line on the Raspberry Pi, though there is external software that can be used for network and node management. In the case of this project, the Z-Ware and ZWeb APIs provided by Silicon Labs were used to create an end-user network management tool. These APIs may also be integrated into the Android Studio app to create a more streamlined user experience.

3.2.5. Connection to the Internet

The hub contains an ethernet interface, so as long as the device is plugged into a router the system will be able to interact with the internet. However, the router will require a DHCP server to allow the hub to have permission to use the connection. This can be done by the user through a connected computer and by editing the router settings through its provided settings program.

The hub will need to have the specific URL of the database it is trying to connect to, and will then create an instance of the database in the code. Once this instance is established, the hub will have the ability to send read/write commands to the database, as well as collect user commands stored in the database. An example code structure can be seen below:

```
Pseudocode for Connection with Internet

#include firebase library

void check_connection{
        //Pings website to determine if internet is connected
        //If ping fails, returns false and alerts the user

}

void loop(){
        //run check_connection
        //If connection exists
                //Utilize firebase library to initialize connection
                //Reads/Writes database if request is added
}
```

Figure 3.6: Internet Connection Pseudo Code

3.3 Data Storage

The data collected from the hub will be written to a Realtime Firebase Database. This database is a free system provided by Google for use with prototyping and app storage. This system provides a way for the app and the hub to communicate and share data or commands over the internet. The database will store the user information for the account, as well as the data collected from each of the outlets. The information in the database is stored in the form of JSON trees, which retain information with a branch system. The JSON tree structure for the WiSSP Firebase Database, in the form of a flow chart, can be seen below:

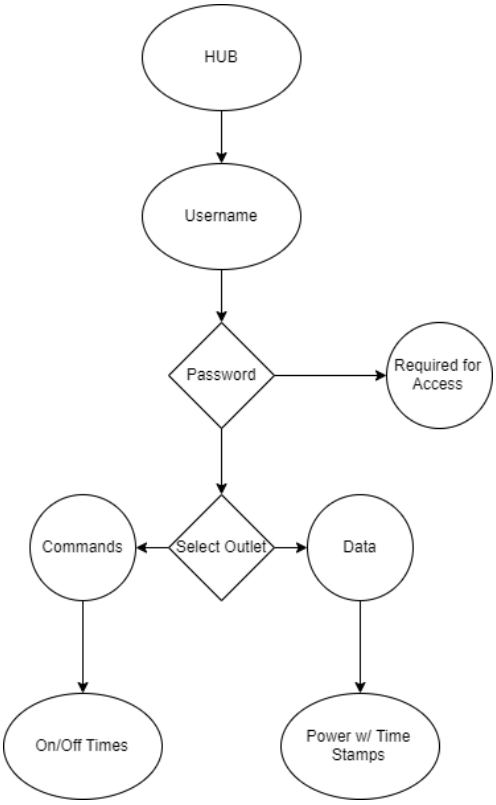

Figure 3.7: Database FlowChart

3.4 User Interface

Android Studio will be used to create the user interface. The program provides a mix of drag and drop design elements for easy visual manipulation, as well as Java code to edit the functions and features the application can provide. The app employs an account system, and allows the user to create an account to keep track of their hub systems.

Figure 3.8: WiSSP Login Screen Example

The app will collect the data stored in the Firebase Database, and will be able to display the information in a simple fashion where the user can easily interpret the data. It will also have the ability to selectively view data based on date.

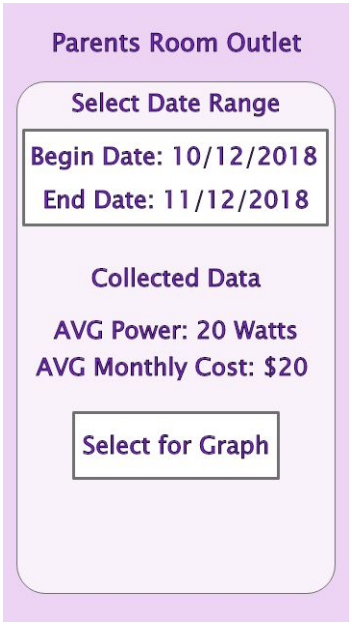

Figure 3.9: WiSSP Outlet Data Screen Example

The app allows the user to perform various tasks for the outlet as well, such as scheduling an on and off time for the outlet to follow, and allows the user to add additional hubs to the app for better management of the system. For additional security, the application will also allow the user to set a PIN passcode, as this will prevent unauthorized users from manipulating the system.

Figure 3.10: WiSSP Outlet Security Code Example

4. Testing and Design Changes

4.1. Hardware

Prior to building the circuit described above, the components were tested individually to ensure they functioned as expected for the design. When appropriate tests and changes were complete, the circuit was constructed and testing was performed along the way. The circuit was first constructed on a prototyping board and tested. Then a second prototype with new components was constructed on a printed circuit board. This section describes both prototypes.

4.1.1. Prototype One

Individual Components

Voltage Dividers

In the circuit, there are two voltage dividers. One is used to divide the value of the DC supply voltage in half in order to be referenced by the microcontroller for the current calculation. The other is to divide the AC line voltage by approximately 50 in order to be measured by the microcontroller. The voltage dividers were characterized by applying a range of test voltages and measuring the output. The one-half divider was implemented using two 1MΩ resistors. The following equation was determined through testing. The test results can be found in Appendix B.

$$DC\ Supply\ Voltage = Divider\ Output * 2.09666 \quad (1)$$

The values of voltage divider for the line voltage were determined such that a very small amount of current would flow through it. Additionally, the peak of the output should not exceed 3.3V, as that is the max input voltage for the microcontroller. The values of 1.075MΩ (using a 1M and a 75k in series) and 20kΩ were picked. The following equation was determined through testing. The test results can be found in appendix B.

$$AC\ Line\ Voltage = Divider\ Output * 55.342 \quad (2)$$

Current Sensor

The ACS723 bilateral current sensor is used to measure the current passing through the load. In order to test this device, a DC power supply was used as a current source. The current was measured using an Agilent DMM as well as the ACS723. The signal on the output is in volts and is supposed to follow the following relationship—voltage out is equal to half the supply voltage plus 0.1 times the current passing through the device. The device was tested to determine how true this relationship was. It was determined that there was a slight offset in the relationship and the actual equation is the following, and the results can be found in appendix B.

$$Load\ Current = [Signal\ Voltage - (\tfrac{1}{2} * V_{supply} + 0.005)] \div 0.1 \qquad (3)$$

This equation was tested with variances in supply voltage as well, and the percent error was determined to be within 0.22%.

Relay Module

The relay circuit shown in Figure 3.3 was built and tested to determine functionality and current consumption. The 5V supply and 3.3V signal were provided by a DC power supply and the current sourced by each was measured. The current sourced by the 3.3V signal was 1.25mA and the current sourced by the 5V supply (which charges the relay) was 161.5mA. These values fall well within the limits of the microcontroller and AC/DC converter.

AC/DC Converter

The AC/DC converter was tested by simply hooking up the AC voltage to the appropriate pins and testing the output voltage and also measuring the voltage of the output with a load. The relay coil was used as the load. In both cases, the output voltage was measured at 5.02V which is well within the tolerance of the expected 5V.

OP-Amp

The OP-Amp was tested for linearity by configuring it as a unity gain buffer and half wave rectifier, inputting a signal and measuring the output on an oscilloscope. Strange behavior was occurring when the signal voltage was near VCC. The output would jump to VCC when the input was around 75% of VCC. When the input is below that threshold, the amplifier behaves linearly. This behavior can be seen in appendix B. Since the input of the microcontroller can not take more than 3.3V, which is below the threshold where the OP-Amp behaves strangely, we decided to move forward with this OP-Amp.

System Level Testing

Once all the components were determined to be working properly, the system was constructed following the schematic shown Figure A1 in Appendix A. Upon construction of the circuit, the functionality was tested and all substructures were powered through the AC voltage and the AC/DC converter. A 3.3V signal was used to trigger the relay, which worked as expected. The outputs of the system (Two voltage dividers and current measurement) were tested. The outputs were extremely close to the expected values as shown below in table 4.1.

Table 4.1: System Test Measurements

	Expected	Measured
Vdc	5.02	5.02
Current output	2.51V	2.503V
AC divider peak	3.15V	3.16V
Vcc (DC) divider	2.402V	2.401V

4.1.2. Prototype Two (Printed Circuit Board)

After the testing of the first prototype, a printed circuit board was designed to contain the circuit and new components were ordered, shown in Figure 4.1. The new components were individually tested and then soldered onto the PCB for system level testing and interfacing with the microcontroller. Due to a change in microcontroller being used, some signal values had to be

adjusted. The ADC on the new microcontroller can only take signals up to 3.08 volts. The decision to change microcontrollers will be discussed in section 4.4 and the individual changes in signals will be discussed under “Individual Components” in this section. The updated schematic can be found in Figure A2 in Appendix A.

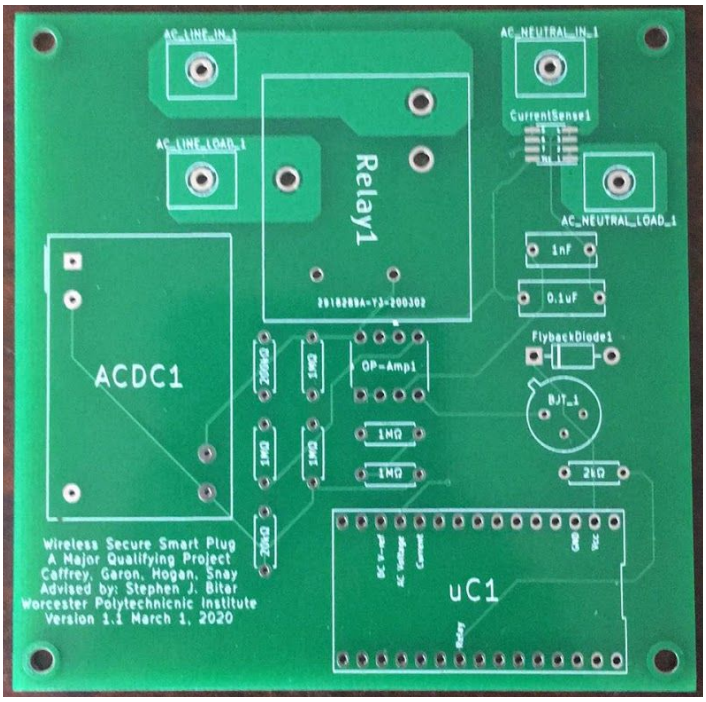

Figure 4.1: Printed Circuit Board

Individual Components

Voltage Dividers

The voltage divider used to divide the DC supply voltage in half to be referenced by the microcontroller was not changed, however it was characterized again since new resistors were used and tolerances exist. The voltage divider to measure the AC line voltage was adjusted to divide the line voltage by around 60 to comply with the input constraints of the new microcontroller. Additionally, a new voltage divider to divide the current signal in half was added so that the current signal is within the constraints of the new microcontroller.

The voltage divider to cut the DC supply in half was again implemented with two 1MΩ resistors. The following equation was determined through testing. The test results can be found in Appendix B.

$$DC\ Supply\ Voltage = Divider\ Output * 2.106375 \quad (4)$$

The values of the voltage divider for the line voltage were adjusted so that the signal would stay below the 3.08V maximum for the microcontroller. The values of 1.2MΩ (1MΩ and 200kΩ in series) and 20kΩ were used. The following equation was determined through testing. The test results can be found in Appendix B.

$$AC\ Line\ Voltage = Divider\ Output * 61.2451 \quad (5)$$

The voltage divider to cut the current signal in half was implemented with two 1MΩ resistors. The following equation was determined through testing. The test results can be found in Appendix B.

$$Current\ Sensor\ Output = Divider\ Output * 2.092923 \quad (6)$$

Current Sensor

Another ACS723 current sensor was used to measure the current passing through the load. For prototype two, however, a unilateral ACS723 was used, as it was not necessary to measure current below zero. The new current sensor was tested in the same way as the one used in prototype one. A DC power supply was used as a current source and the current was measured using a digital multimeter as well as the ACS 723 and the results were compared. For the unilateral sensor, the following relationship is supposed to be followed—voltage out is equal to one tenth of supply voltage plus 0.2 times the current passing through the device. The relationship was tested. It was determined that there was a slight difference in the zero current

voltage (0.10123*Vcc instead of 0.1*Vcc) and the sensitivity was slightly different. The following relationship was found and the results can be found in Appendix B.

$$Current = [Signal\ Voltage - (V_{supply} * 0.10123)] * 5.181647 \quad (7)$$

Combining the above current equation with the equation for the voltage divider, the following full current voltage relationship was determined.

$$Current = [(Signal\ Voltage * 2.092923) - (V_{supply} * 0.10123)] * 5.181647 \quad (8)$$

Relay Module

The relay subcircuit was not changed from prototype one. New parts were used so it was tested to ensure proper behavior. The 5V supply and 3.3V signal were provided by a DC power supply and the current being sourced by each was measured. The current sourced by the 3.3V signal was 1.3mA and the current sourced by the 5V supply (which charges the relay) was 159mA. These values fall well within the limits of the microcontroller and AC/DC converter.

AC/DC Converter

The AC/DC converter was tested in the same way as prototype one. It was connected to the 120V AC line and the DC output was tested with and without a load (the load used was the relay coil). In both cases, the output voltage was measured to be 5.01V which is within tolerance of the expected 5V.

OP-Amp

The same model OP-Amp was used as in prototype one and was tested in a similar manner. The amplifier was connected as a unity gain buffer and half wave rectifier. The same behavior was observed. When the signal voltage approached 75% of VCC, the output would jump to VCC. However, since the signals being input into the amplifier are at or below 3.08V which falls below 75% of VCC, this behavior will not affect the performance of the device.

Graphs of the input and output of the OP-AMP with VCC = 5V and signal voltage equal to 5V and 3.3V can be found in Appendix B.

System Level Testing

Once all of the components were determined to be working properly, the system was constructed according to the schematic shown in Figure A2 in Appendix A. Upon construction of the circuit, the functionality was tested and all substructures were powered through the AC voltage. A 3.3V signal was used to trigger the relay which worked as expected. The outputs of the system (Two voltage dividers and full current equation) were tested. The outputs measured were extremely close to expected and are shown below in table 4.2.

Table 4.2: System Test Measurements

	Expected	Measured
Vcc (DC)	5.01	5.01
Zero Current Output (DC)	0.255	0.261
AC divider peak	2.76	2.79
Vcc (DC) Divider	2.386	2.386

4.2. Software

In the process of building the WiSSP application and coding the functionality of the outlet system, some errors were discovered and some code was found to be ineffective. Due to time constraints some of these features had to be eliminated.Changes that occurred will be discussed in this section.

4.2.1. Application Changes

The android studio app was a straightforward endeavor for the majority of the project duration. Most of what was predicted to be included in the application was able to be designed, although some decisions had to be made to create a more cohesive system.

Firebase Syntax

The Firebase database remained relatively the same structure as specified. The only changes made in the overall structure were focused on semantics, such as the labeling for each outlet. The database orders numbers and characters lexicographically, which means its ordered by the first character to appear and then by the order in the alphabet or numerically. For example, the phrase “A01” and “A02” would be put in this order. This became a problem because of how the outlet keys were created. Whenever an outlet is added to the database the outlet used to be called “Outlet_#”, where the # was indicative of its place in the database. For example, the intention was that Outlet_9 would be the ninth to be added. The issue arises once the tenth outlet is added, where the database notices the 1 at the same spot in the name as the 9, so 10 will be placed after Outlet_1 instead. To solve this issue, placeholders were created for unused decimal places, which allowed the outlets to be arranged numerically without concern.

Web Requests

The original intention for the way the app would communicate with the hub was to be through the Firebase database system. Although the system would work, the method the hub would collect the commands with proved to be inefficient. To have the database be the

intermediary, the hub would constantly be checking the database for a change in each key, and if one was found the related action would occur. The issue was that it made the system slower and used more power than was worth, so the solution was to create a system that would send web requests from the app to the hub whenever certain actions occurred, such as turning the outlet on and off.

User Login/Outlet PIN

Due to both timing constraints and the lack of a true purpose for implementing, it was decided to eliminate the use of both a user login option and PIN locks. It is a very simple code to implement, and won't affect the prototype in any way because we can't demonstrate adding multiple users. At a future date it would be very simple to update the app to complete this if this was to become a commercial product.

4.3. Wireless Communication

4.3.1 Testing and Re-Design

Following the steps outlined in Section 4.2.1 of "Z-Wave 700 Getting Started for Controller Devices," the Silicon Labs UZB Stick was used with the Raspberry Pi to create the Z-Wave Controller. This involved the use of the ZWave SDK available on the Silicon Labs website. The Silicon Labs Z-Wave Development Kit was used to create an End Device Node. Following the steps outlined in "Z-Wave 700 Getting Started for Controller Devices," the Binary Switch Sample Application was implemented and added to the network. The Z-Ware Web API was used as a network manager to add, remove, and control nodes. Using this software, the network and sample application were both verified to be functional.

Upon further examination, it was determined that the Z-Wave Development Kit was not an optimal way to implement a Z-Wave End Device. The supplied sample applications were large and difficult to understand due to the complex nature of the WSTK Development Board itself. We were unable to determine which parts of the sample application were necessary for the functionality of the app and which were not. Due to this, the Z-Wave Development Kit was

determined to be unusable for the purposes of this project. In its place, an Arduino-based microcontroller known as the Z-Uno was used. This microcontroller operates like a regular Arduino microcontroller, but has a Z-wave chip already attached and integrated. Generic sample applications were provided and implemented, as described in the next section.

The Silicon Labs UZB Stick and Z-Ware Web API were also abandoned. The Z-Ware Web API, while fine for testing, did not allow for any user customization or interaction. We were unable to determine a simple way to use this API to accomplish the goals of our project, such as turning the outlet on and off via command from the mobile app or logging the Voltage and Current readings from the outlet. As other software options were explored, it became apparent that the Silicon Labs UZB Stick, a Bridge Controller, was not compatible with a vast majority of user-oriented solutions. For these reasons, the Silicon Labs UZB Stick and Z-Ware Web API were determined unusable in the final design of this project. The UZB Stick was replaced by the Zooz Z-Wave USB Stick, a Static Controller. The Web API was replaced with an IOT device management software known as Home Assistant.

4.3.2. Final Design

Home Assistant is an open-source software that allows the user to control all of their IOT devices in one place. It is frequented by DIY enthusiasts, and therefore is very customizable, re-programmable, and intuitive. Home Assistant was installed using the “Manual Install” instructions as opposed to the all-in-one Hass.io Raspberry Pi Installation. The manual install was chosen as the latter installation method does not allow the user easy access to the rest of the operating system once installed, meaning that all work would have to be done through Home Assistant. The Zooz Stick was installed using the Z-Wave Integration in the Configuration menu, as per the Z-Wave instructions on the Home Assistant website. The Z-Uno was then paired using the “Add Node” button on the Z-Wave Control Panel.

Home Assistant Automations allow for a set of actions to take place whenever a certain condition is met or trigger occurs. In the case of this project, most actions were triggered via a Web Request. Actions such as adding an outlet, turning an outlet on or off, and implementing scheduling for an outlet were all accomplished using Automations. One problem with the Web

Request system, however, is that Home Assistant does not accept HTTPS web requests, only HTTP. Because of this, a "middel-man" server was hosted on the Raspberry Pi which accepted HTTPS requests from the mobile application, and then sent HTTP requests to Home Assistant.

The final design implements and manages Z-Wave through a combination of Home Assistant Automations and Python Scripts. The user begins by adding an outlet to the system. They can do so through the mobile application, which then sends a Web Request to a Python Server hosten on the hub. This server will then send a Web Request to Home Assistant to trigger the "Add Node" process. Once the node has been successfully added, a Python script will run to add the node to the database and implement scheduling for that node in Home Assistant. The user can then turn the outlet on or off or change the scheduling of the outlet, both of which will send a Web Request to the Hub. The hub will then take the necessary actions to perform the requested action, and then update the database accordingly. Alongside the servers, another Python script runs continuously, checking the Z-Wave log every minute for Current and Voltage readings from the Z-Uno and uploading them to the database.

The Web Request system is designed to make the hub operable even without the internet for extended periods of time. The scheduling is implemented on-device through Home Assistant, meaning it will work even when the Internet is disconnected. Similarly, the Z-Wave communication will still function without the internet, meaning log values of Voltage and Current can still be read. The only features that do not work without the internet are those that rely on user input from the app, such as adding a node or turning an outlet on or off on command. Since these requests would have to come from the app anyway, it does not matter if they cannot be performed without internet connection as the request cannot be made in the first place.

4.4. Microcontroller

From the initial design, we planned to use the Teensy 4.0 microcontroller connected with the Z-Wave transmitter chip provided with the Silicon Labs development kit. However, the chip required more processing power and peripheral circuitry than can be provided by the Teensy. Instead, we decided to use an Arduino-based microcontroller called the Z-Uno. This device can be formatted to act as up to 32 different Z-Wave devices at a time, allowing it to transmit the

voltage and current values and control the circuit. With this change, the code became a different structure than the pseudocode originally stated. Most of the transmission via Z-Wave is handled by the Z-Uno's internal functions so the only code to design manually is the logic to calculate the voltage and current values. The code for the internal macros was written based on the example code for an analog temperature sensor input. Other than our pin values, however, none of the changes will affect the rest of our design. All of the analog sensors and inputs can still be registered on this controller. To ensure the controller was reading and interpreting values properly, the internal ADC needed to be calibrated.

4.5. ADC Calibration

In order to ensure the proper sensor readings from the Z-Uno, the device's internal ADC needed to be characterized. According to the documentation, the Z-Uno's internal ADC can use up to a 12 bit resolution, distributed between ground and the controller's operating V_{CC}, which is about 3.3 V. In order to ensure our calculations were correct for the final sensor values, however, this had to be verified. Using the lab DC voltage supply, one voltage channel was connected to the ADC and another independent channel was connected to the board V_{CC}. The V_{CC} was powered with 5 volts, just as the outlet design will in practice. The ADC channel was then increased by increments of 10 mV, with the Z-Uno outputting the ADC values it was receiving. The goal was to find the exact voltage value at which the ADC produced 4095, which is the highest value produced at 2^{12} resolution. Nominally, this would be when the channel reached 3.3 V. After testing, however, it was determined that the ADC actually produces maximum value at 3.13 V. This test procedure was repeated for the three ADCs used in the final design in order to ensure that they all had the same maximum value; the tests confirmed that this was the case. Thus, the ADC could properly register any input from 0 V to 3.13 V, with the calculation for the actual value received being 3.13/4096.

4.6. Subsystem Interface

4.6.1. Sending Values to Hub

When interfacing the PCB and the Zuno microcontroller and sending the measured values to the hub, the values for voltage were very accurate (within plus or minus 1V), however the

values for current were not accurate. It was determined through testing the current signal with an oscilloscope that there was a significant amount of noise on the signal. Ideas for how to deal with the noise were considered, however due to time constraints and the inability to redesign the board at this stage, a coding solution was explored. Since the measurement for the current signal is taken as the peak of the signal and the peak of the noise is consistently around 400mV, code on the microcontroller was written to subtract 400mV from the signal. This isn't an ideal solution and will be further discussed in Section 5.5.

5. Results and Recommendations

5.1. Signal Noise

As discussed in the previous section, there was a significant amount of noise on the signals. This especially affected the current measurement signal. The solution that was used was to simply subtract the typical noise from the wave. This is not ideal as it does not account for variances in the noise. For future development, we recommend taking a different approach for reducing signal noise. In this design, a capacitor was placed between VCC and ground as a lowpass filter, however further implementation of filters on each signal could reduce noise further including a focus on where the capacitors are located on the physical circuit board.

It appeared that a majority of the noise was injected from components in the system, most likely from the AC/DC converter and the Relay. Since the current sensor that is used gives an output which relies on supply voltage in the calculation, any noise on the supply voltage is propagated to the current signal. A stable voltage regulator could be used to ensure that the supply voltage does not vary. Potentially, a 12V power supply could be used to power the relay coil and some sub-circuitry and a 5V regulator (powered from the 12V) could be used to supply a stable reference voltage. Additionally, a current sensor whose output does not rely on supply voltage could be used.

It was observed that a level of 60Hz noise was being produced around the relay and high frequency noise was being produced from the AC/DC converter. This was measured by grounding the oscilloscope ground to the ground of the circuit and moving the tip of the probe close to certain components. Better shielded components could reduce the noise that propagates to the rest of the circuit. A ground plane, as opposed to traces, could further reduce noise on the signal traces.

5.2. Physical Design

The system was designed with physical footprint in mind. The intention is that the device would be implemented behind a large appliance that is pushed against the wall. Ideally, the device would plug into the wall and not stick out very much. This ideal was not accomplished due to focus on the circuitry and lack of focus on the physical shape. In order to still be used for

an appliance, the design features a box with a cord to the wall outlet and another cord to the appliance or device being controlled and monitored (shown in Figure 5.1). Future development of this device should pay special attention to the physical space taken up by the product. Changes that could be made to reduce size include selecting a relay with a smaller footprint and 3D printing an encasement for the device.

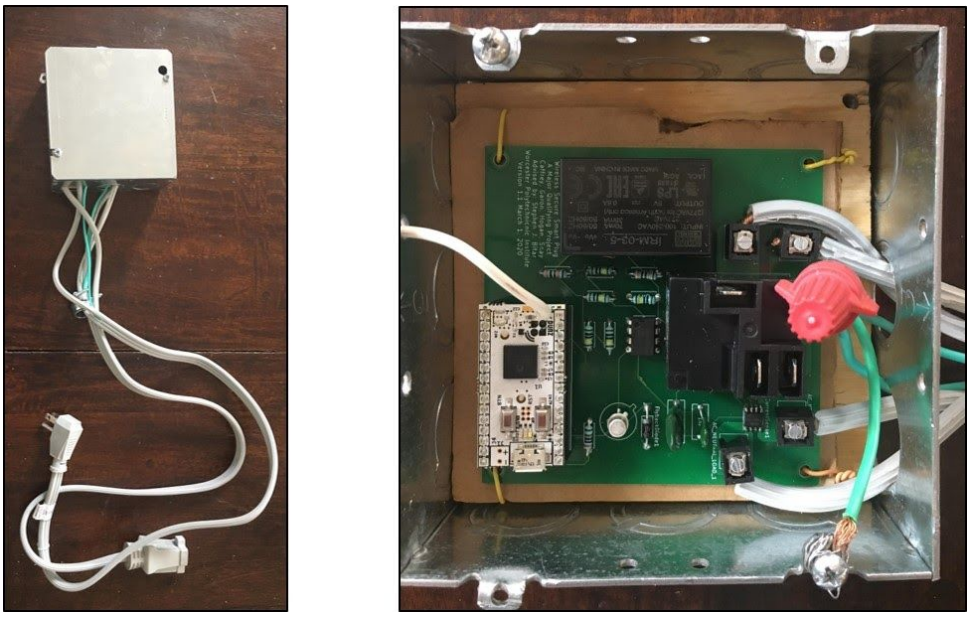

Figure 5.1 Outlet Device

5.3. Isolation of AC and DC

In the design that was used, the DC ground and the AC neutral had to be tied together in order to measure the voltage using the voltage divider. This is not the safest design since there exists a path for an electron from the AC supply to get to the DC circuitry. A safer design would include complete isolation from the AC supply. Galvanic isolation exists in the power supply since the power is converted via a magnetic field. An isolation amplifier (optical or otherwise) could be used to isolate the branches.

5.4. Relay Selection

The relay that was used is a normally connected relay. In future development of the product, it would be beneficial to select a latching relay. In the current design, in order for the

device to power off what is plugged into it, the relay must continuously be charged which is a waste of power. A latching relay would allow for on/off control that does not waste power as the relay coil only needs to be charged for a moment to latch from on to off or vice versa.

5.5. Android Application Security Risks

An issue was encountered when trying to send a web request from the android application to the raspberry pi hub. The issue was related to the android studio software: when trying to send a web request to a server without a trusted certificate the program will choose to block that command automatically. Because the server isn't a final product, it had no trusted certificate that android studio would allow. To circumvent the android firewall an object in the code, called a trust manager, was added to allow communication. The trust manager retains the settings and preferences for when to connect to a server and when it is not safe. Unfortunately to create this connection with the server the trust manager is coded to allow any web requests, which is extremely unprotected. This was decided to be a fine solution for testing purposes, but for a final product this will need to be changed. In a future endeavor, the server could be one of higher quality, or it could be given the trust certificate it requires. Another solution is to create a trust manager in the app that specifically trusts only the IP address of the hub. In either case the communication would be more secure and would allow for the use of web requests.

5.5.1 Account Support

For the prototype the ability to add a specific user account is no longer present. In the future it would be very ideal to allow for a situation where the user could create a profile and control their own settings. This could be done in firebase by using the cloud server rather than the real time database. This allows for functions to be carried out in the database and may also create user specific profiles for each of the hubs. The feature would also allow for better app security, as a password would be required for using the application.

5.6. Microcontroller and ADC

For the microcontroller, the component itself is very well designed for a DIY Z-Wave device. The controller connects well with the Z-wave gateway run from the Raspberry Pi and formats the data well when reading the current and voltage. It also leaves little to no delay time when accepting the command from the app to turn the circuit on or off. The main downside to the Z-Uno with our project is the ADCs. The internal ADCs have a 12 bit maximum resolution, but they have a difficult time with reading an external reference. This downside forces the design to use the controller's internal voltage reference, which floats anywhere from 3.13 to 3.24 volts when being powered by 5 volts. This inconsistency is not detrimental for the average person, but increasing this accuracy will improve the device as a whole. There are two possible solutions that would solve this problem. The first is using three external ADC to read the circuit values. This allows the integration of an ADC with better accuracy into the circuit. This new ADC can also have better resolution and a higher voltage reference. With a 5 or 7 volt ADC, some of the voltage dividers introduced in the circuit could be removed. Doing so would additionally have the benefit of removing some internal noise.

The second solution is using a regulated external voltage reference and putting that into the Z-Uno's pin. The 5 volt DC voltage in the current circuit is not consistent enough to use for a voltage reference, hence the use of the internal reference. If a regulated voltage was used, the Z-Uno could use the built-in ADCs more effectively and accurately, while also reducing the need for some of the voltage dividers and calculations. This will make the voltage and current measurements much more accurate and thus improve user data and the device's usefulness overall.

5.7. Wireless Communication

The implementation of the Z-wave management software on the hub features Web Requests as its primary method of communication, and involves both the Home Assistant Web Server as well as a locally hosted Python Web Server. This implementation has both advantages and disadvantages. One pro is that, since the Python Web Server is part of the communication

between the mobile application and Z-Wave software, the Python scripts can always "know" what is happening and respond accordingly. If this were not the case, Home Assistant itself would have to trigger any other scripts necessary to perform an action, which becomes more convoluted. However, a con of this implementation is that a Web Server must be constantly hosted on the Raspberry Pi, which is already constantly running Home Assistant and other scripts. This makes the implementation not as efficient as it could be.

There are some features that we hoped to implement but did not have time. Firstly, the Web Request system depends on the Hub's IP address being accurately updated in the database. Currently this must be done manually. Secondly, the power monitoring functionality, though it can operate without the internet, does not have any procedure to save values on the device if they cannot be uploaded to the database. A system should be implemented where these values are saved and uploaded once internet connection is restored. Finally, there is no code-procedure in general to deal with Internet disconnection. For example, if connection to the database cannot be made, an error will be thrown and the program will end. None of these issues are difficult to resolve, but rather they simply are not necessary in a proof-of-concept prototype. Regardless, any continuation of this project should address these concerns in some way.

6. Conclusion

Overall the project satisfies what it was designed to do. It consists of an outlet, a hub, and an app. The outlet measures power which can be tracked on the app and the user can turn the outlet on and off from the app. There is plenty of room for improvement before this product can be brought to the market, but it serves as a stepping stone for a smart outlet unlike the others available today. Some of the areas that could use the most improvement include reduction of noise on the signals, better implementation of the analog to digital conversion, more security between the hub and app, and more that have been discussed in the recommendations. Despite the areas that could use improvement, the project was successful in being what it was designed to be.

Works Cited

"Amazon.Com: Customer Reviews: Amazon Smart Plug, Works with Alexa." Web.

"Basics of UART Communication

." *Circuit Basics.* 2017. Web. Nov 11, 2019 <http://www.circuitbasics.com/basics-uart-communication/>.

Benson, Lisa, and Karen Reczek. *A Guide to United States Electrical and Electronic Equipment Compliance Requirements*. U.S Department of Commerce, 2017. Print.

"Best Smart Wifi Outlets and Plugs." *Postscapes.com.* March 19, 2019. Web. September 11, 2019 <https://www.postscapes.com/smart-outlets/>.

Bitar, Steven J. *Meeting Conversation*. Ed. Daniel Caffery, et al., 2019. Print.

Brown, Rich. "The Best Smart Home Devices of 2019." *CNET* (2019)Web. Sep 13, 2019.

Charter, Mark. "How to Set Up a Smart Speaker with Privacy in Mind." *KGUN* (2019)Web. Sep 13, 2019.

Dew, Wendy. "Energy Vampire ." *The EPA Blog.* October 30, 2012. Web. October 13, 2019 <https://blog.epa.gov/2012/10/30/energy-vampire/>.

Federal Communications Comission et Al. "Radio Spectrum Allocation." *Federal Communications Commission.* Web. Jan 16, 2019 <https://www.fcc.gov/engineering-technology/policy-and-rules-division/general/radio-spectrum-allocation>.

Frenzel, Louis E. *Handbook of Serial Communications Interfaces : A Comprehensive Compendium of Serial Digital Input/Output (I/O) Standards*. Oxford, UK: Elsevier Science, 2016. Web.

Gebhart, Andrew. "7 Steps to Control Your Home with Google Assistant." *CNET* (2019)Web. Sep 13,

2019.

"Google Smart Home Devices." Web. <https://store.google.com/category/connected_home?gclid=CjwKCAjwk93rBRBLEiwAcMapUflCoAQuxPinuTD2EyWem3xzNGlM-AZguD1j92DdjRMAyB1IhG3dNBoCY0AQAvD_BwE&gclsrc=aw.ds>.

K. Gill, et al. *A Zigbee-Based Home Automation System*. 55 Vol. , 2009. Web.

Liu, Shanhong. "Global Smart Home Market Revenue 2016-2022." *Statistica.* 8/9/ 2019. Web. 9/9/2019 <https://www.statista.com/statistics/682204/global-smart-home-market-size/>.

Meier, Alan. "Standby Power." *Berkeley Lab.* 2019. Web. September 14, 2019 <http://standby.lbl.gov>.

Meyers, Robert J., H. Scott Matthews, and Eric D. Williams. "Scoping the Potential of Monitoring and Control Technologies to Reduce Energy use in Homes." *Energy and Buildings* 42.5 (2009): 563-9. Web.

Miller, G. E. "Stop Wasting Money on Electricity! Your Guide to Identifying & Unplugging Standby Power Appliances." *20 something finance.* May 5, 2019. Web. September 12, 2019 <https://20somethingfinance.com/electrical-leaking-standby-appliance-list/>.

Mysen, Andreas Greftegreff. *Smart Products: An Introduction for Design Students*. Department of Product DesignPrint.

O'Boyle, Britta. "Best Smart Plugs 2019: Google, Alexa and HomeKit Control." *Pocket-lint* (2019)Web. Sep 13, 2019.

P. m. Linh An, and T. Kim. *A Study of the Z-Wave Protocol: Implementing Your Own Smart Home Gateway*., 2018. Web.

S. J. Danbatta, and A. Varol. *Comparison of Zigbee, Z-Wave, Wi-Fi, and Bluetooth Wireless Technologies used in Home Automation*., 2019. Web.

Silicon Labs. "Z-Wave Global Regions." *Silicon Labs.* 2020. Web. Jan 10, 2020 <https://www.silabs.com/products/wireless/mesh-networking/z-wave/benefits/technology/global-regions?fbclid=IwAR2FA6qgesAowf-1ZbnrC8Lqoj5zjGWFVxQe2a0Edm9MFurjJ-qfruSIW_4>.

"Smart Plug with Energy Monitoring | Currant Smart Outlet." Web. Sep 13, 2019 <https://www.currant.com>.

"What is a Smart Device?" *Technopedia.* Web. 9/9/2019 <https://www.techopedia.com/definition/31463/smart-device>.

Winfrey, Graham. "Why 'Smart' Homes are Creeping Out Consumers." *Inc.com* (2016)Web. Sep 13, 2019.

Wright, Adam, Jitesh Ubrani, and Michael Shirer. *Double-Digit Growth Expected in the Smart Home Market, Says IDC*., 2019. Print.

Appendices

Appendix A: Outlet Schematics

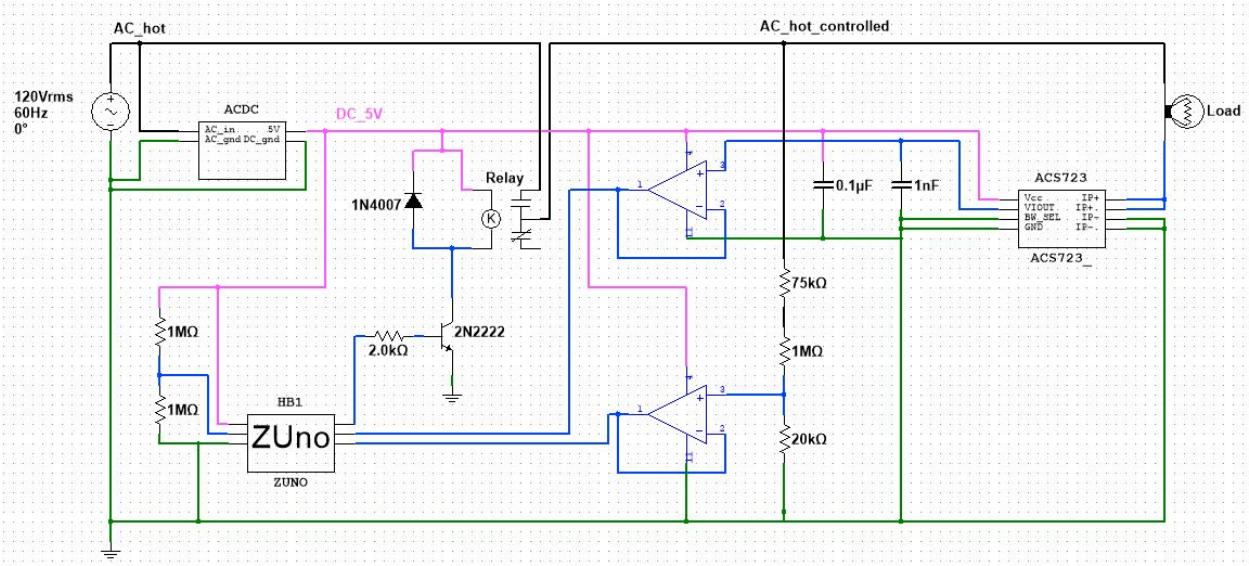

Figure A1: Original Outlet Schematic

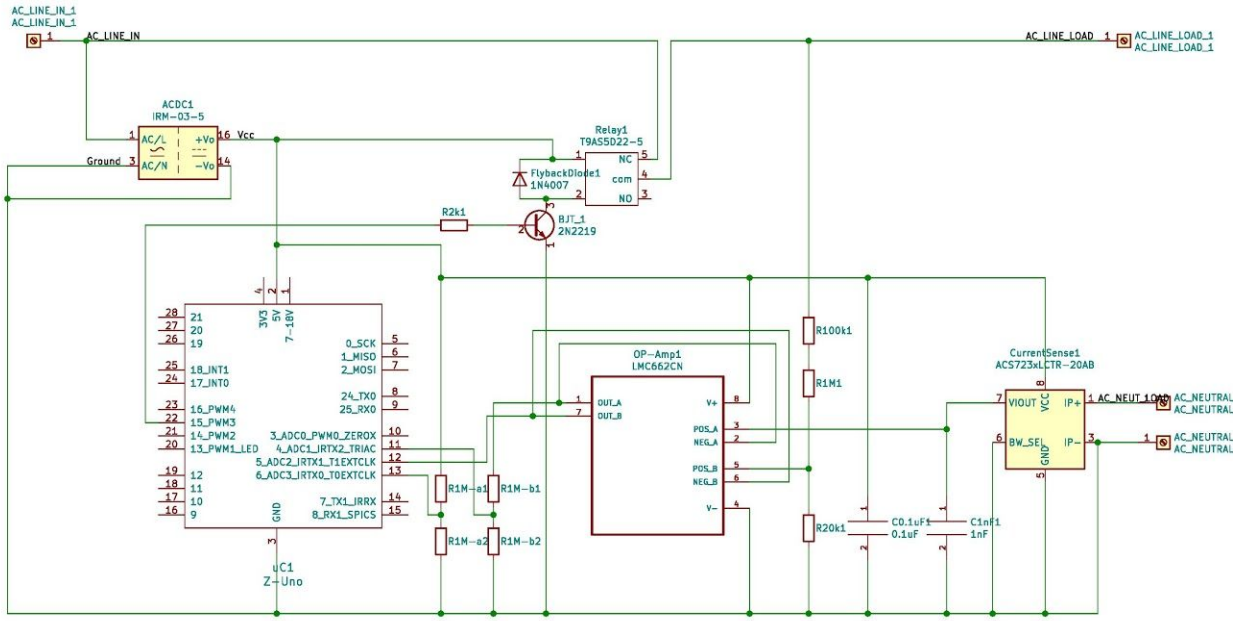

Figure A2: Schematic Updated for Prototype Two

Appendix B: Component Test Results

Prototype One

Voltage Dividers

Table B1: Characterizing Voltage Divider for DC Supply - Prototype One

Vin	**Vout**	**"gain"**
5.038	2.403	0.476975
4.968	2.370	0.477053
4.470	2.132	0.476957
4.534	2.162	0.476842
4.751	2.266	0.476952
4.823	2.300	0.476882
5.198	2.479	0.476914
4.939	2.356	0.477020
	Average:	**0.476949**
	Inverse:	**2.096659**

Table B2: Characterizing Voltage Divider for AC Line - Prototype One

Vin (V)	**Vout (mV)**	**"gain"**
2.84	51.4	0.018099
4.89	88.5	0.018098
7.55	136.5	0.018079
9.76	176.4	0.018074
12.18	220	0.018062
14.78	267	0.018065
16.65	300	0.018018
19.60	354	0.018061
	Average:	**0.01807**
	Inverse:	**55.34165**

Current Sensor

Table B3: Testing to Determine Current Equation - Prototype One

I (A)	Vout (V)	Sensitivity (V/A)	I calculated*	%error	Error (mA)
0.0000	2.511		0.0100	#DIV/0!	10.0
0.1181	2.523	0.102	0.1300	10%	11.9
0.2013	2.531	0.099	0.2100	4%	8.7
0.3162	2.543	0.101	0.3300	4%	13.8
0.4009	2.551	0.100	0.4100	2%	9.1
0.5040	2.561	0.099	0.5100	1%	6.0
0.6100	2.572	0.100	0.6200	2%	10.0
0.7027	2.581	0.100	0.7100	1%	7.3
0.8161	2.592	0.099	0.8200	0%	3.9
0.9027	2.600	0.099	0.9000	0%	-2.7
1.0247	2.613	0.100	1.0300	1%	5.3
1.1508	2.625	0.099	1.1500	0%	-0.8
1.2716	2.637	0.099	1.2700	0%	-1.6
1.4018	2.650	0.099	1.4000	0%	-1.8
1.5560	2.667	0.100	1.5650	1%	9.0
1.7577	2.685	0.099	1.7500	0%	-7.7
1.9993	2.709	0.099	1.9900	0%	-9.3
2.2599	2.736	0.100	2.2600	0%	0.1
2.4927	2.759	0.099	2.4900	0%	-2.7
2.7534	2.785	0.100	2.7500	0%	-3.4
3.0026	2.810	0.100	3.0000	0%	-2.6
3.1028	2.821	0.100	3.1100	0%	7.2
	avg:	avg: **0.100**	*Calculated (using I=(Vout-(1/2)Vcc+0.005)*sensitivity)		

Table B4: Testing Equation Over Changes in Supply Voltage - Prototype One

Vcc (V)	Vout (V)	V expect	V diff	% error
4.502	2.255	2.251	0.0040	0.18%
4.606	2.308	2.303	0.0050	0.22%
4.746	2.378	2.373	0.0050	0.21%
4.834	2.422	2.417	0.0050	0.21%
4.927	2.468	2.4635	0.0045	0.18%
5.023	2.516	2.5115	0.0045	0.18%
5.101	2.556	2.5505	0.0055	0.22%
5.200	2.605	2.6	0.0050	0.19%
5.286	2.648	2.643	0.0050	0.19%
5.444	2.728	2.722	0.0060	0.22%

avg: **0.0050**

OP-Amp

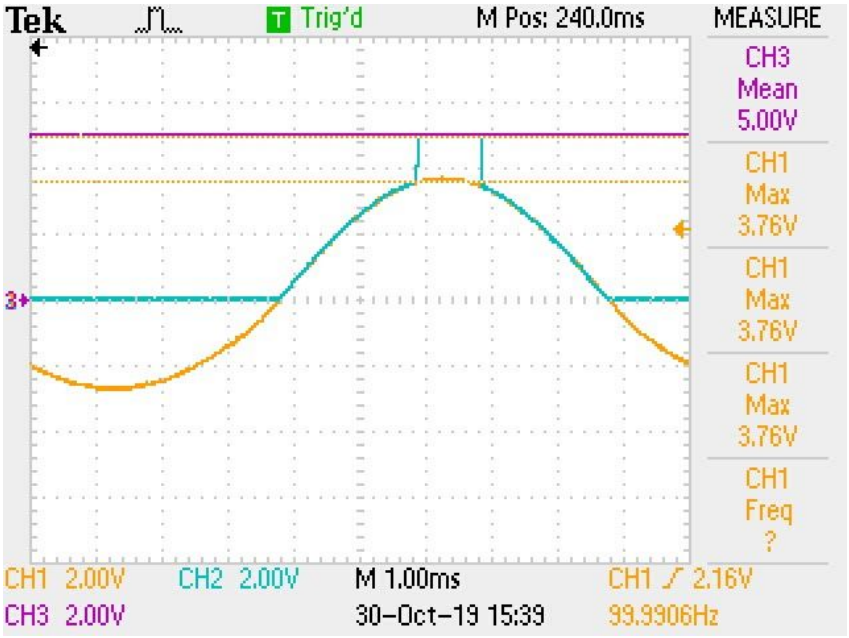

Figure B1: OP-Amp Behavior - Prototype One

Prototype Two

Voltage Dividers

Table B5: Characterizing Voltage Divider for DC Supply - Prototype Two

Vin	**Vout**	**"gain"**
0.9336	0.44333	0.474861
2.5195	1.19604	0.474713
3.9944	1.8963	0.474740
5.2435	2.4892	0.474721
6.1185	2.9045	0.474708
6.8787	3.2654	0.474712
7.8001	3.703	0.474738
8.3418	3.9603	0.474754
9.0755	4.3087	0.474762
10.1128	4.8014	0.474784
	Average:	**0.474749**
	Inverse:	**2.106375**

Table B6: Characterizing Voltage Divider for AC Line - Prototype Two

Vin (V)	**Vout (mV)**	**"gain"**
1.11252	18.166	0.016329
3.0568	49.909	0.016327
5.0315	82.157	0.016329
6.9912	114.15	0.016328
9.2701	151.35	0.016327
11.128	181.7	0.016328
12.858	209.93	0.016327
15.04	245.55	0.016328
17.08	278.92	0.016329
20.067	327.65	0.016328
	Average:	**0.016328**
	Inverse:	**61.2451**

Table B7: Characterizing Voltage Divider for Current Signal - Prototype Two

Vin	**Vout**	**"gain"**
1.08897	0.52042	0.477901
2.3614	1.12833	0.477822
3.4538	1.65	0.477735
4.2845	2.0469	0.477745
4.8483	2.3164	0.477776
5.9455	2.8407	0.47779
6.9504	3.3208	0.477785
7.81	3.7329	0.477799
8.78	4.1962	0.477824
9.6949	4.6325	0.477829
	Average:	**0.477801**
	Inverse:	**2.092923**

Current Sensor

Table B8: Testing to Determine Current Equation - Prototype Two

I (A)	Vout (V)	Sensitivity (V/A)	I calculated*	%error	Error (mA)
0.0000	0.50840		-0.0038	#DIV/0!	-3.7627152
0.1176	0.53070	0.1896	0.1118	-5%	-5.8119841
0.2278	0.55190	0.1910	0.2216	-3%	-6.1610647
0.3878	0.58290	0.1921	0.3823	-1%	-5.5300034
0.5048	0.60550	0.1924	0.4994	-1%	-5.4247781
0.5986	0.62370	0.1926	0.5937	-1%	-4.9188001
0.7859	0.65980	0.1926	0.7807	-1%	-5.1613384
0.9053	0.68280	0.1926	0.8999	-1%	-5.3834542
1.0389	0.70890	0.1930	1.0352	0%	-3.7424639
1.2055	0.74120	0.1931	1.2025	0%	-2.9752613
1.3388	0.76710	0.1932	1.3367	0%	-2.0706004
1.4917	0.79670	0.1933	1.4901	0%	-1.5938451
1.5892	0.81550	0.1932	1.5875	0%	-1.6788789
1.6944	0.83590	0.1933	1.6932	0%	-1.1732773
1.7380	0.84450	0.1934	1.7378	0%	-0.2111119
1.8837	0.87220	0.1931	1.8813	0%	-2.3794862
2.0207	0.89890	0.1932	2.0197	0%	-1.0295076
2.1393	0.92210	0.1934	2.1399	0%	0.58470607
2.2721	0.94760	0.1933	2.2720	0%	-0.0832919
2.3798	0.96830	0.1933	2.3793	0%	-0.5231961
2.5128	0.99410	0.1933	2.5130	0%	0.16330006
2.6277	1.01670	0.1934	2.6301	0%	2.3685254
2.7528	1.04070	0.1934	2.7544	0%	1.62805673

2.8672	1.06250	0.1933	2.8674	0%	0.18796435
3.1366	1.11450	0.1932	3.1368	0%	0.23361557
	avg:	avg:**0.1930**	*Calculated (using I=(Vout-0.10123*Vcc)*sensitivity)		

Table B9: Testing Equation Over Changes in Supply Voltage - Prototype Two

Vcc (V)	**Vout (V)**	**V expect**	**V diff**	**Vout/Vcc**	**% error**
4.6277	0.4682	0.46277	0.0054	0.10117	1.17%
4.7059	0.4761	0.47059	0.0055	0.10117	1.17%
4.8349	0.4892	0.48349	0.0057	0.10118	1.18%
4.9541	0.5015	0.49541	0.0061	0.10123	1.23%
4.9857	0.5047	0.49857	0.0061	0.10123	1.23%
5.00003	0.5062	0.500003	0.0062	0.10124	1.24%
5.0291	0.5091	0.50291	0.0062	0.10123	1.23%
5.1000	0.5165	0.51	0.0065	0.10127	1.27%
5.1691	0.5236	0.51691	0.0067	0.10129	1.29%
5.208	0.5276	0.5208	0.0068	0.10131	1.31%
			avg: **0.0100**	**0.10123**	

OP-Amp

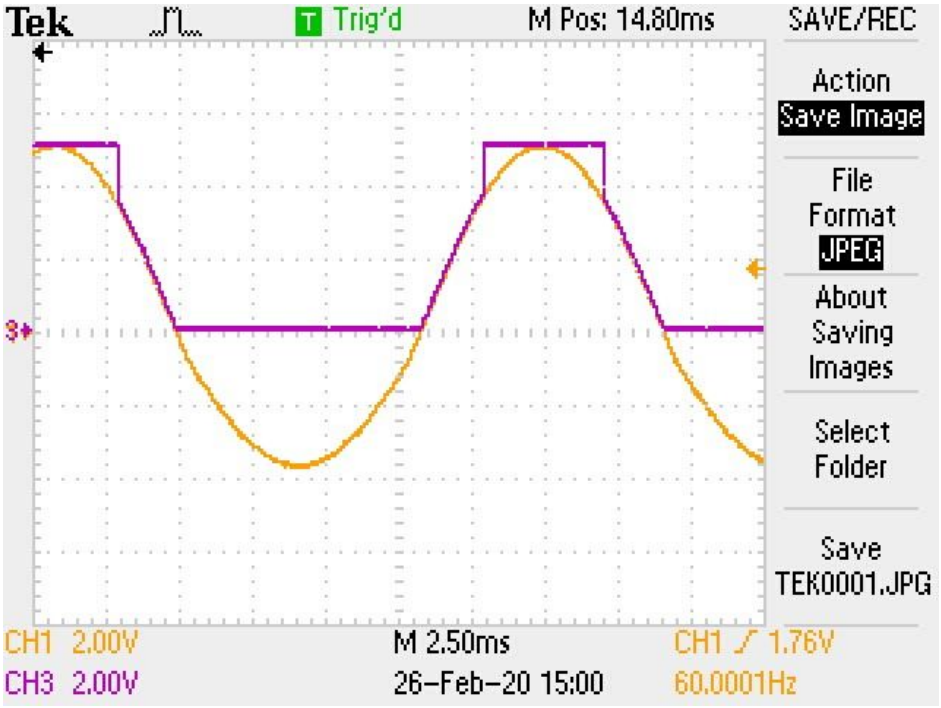

Figure B2: OP-Amp Behavior with Vin = 5V - Prototype Two

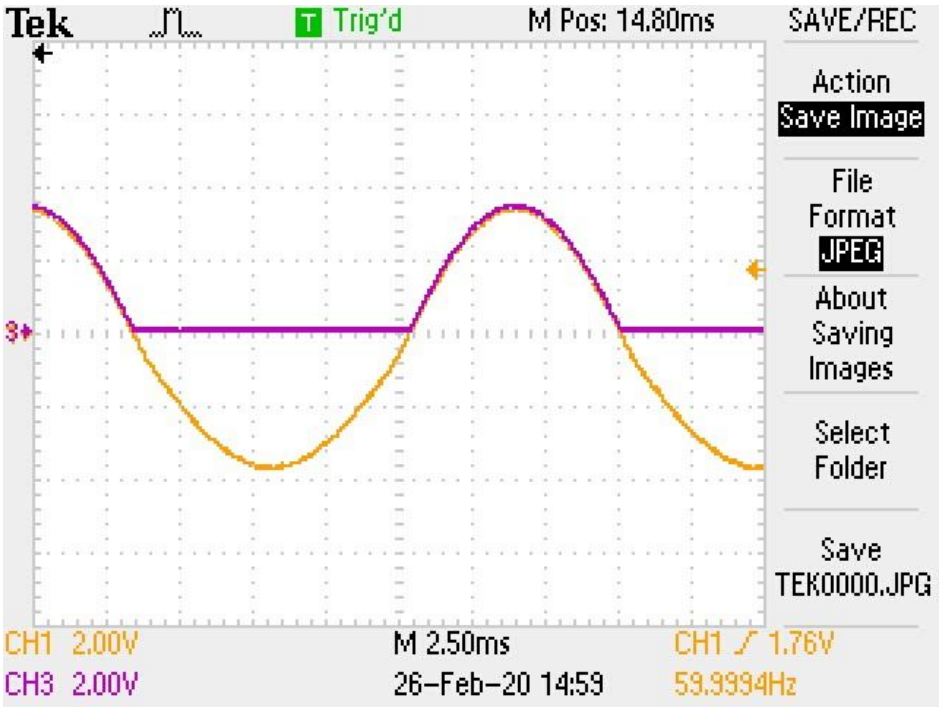

Figure B3: OP-Amp Behavior with Vin = 3.3V - Prototype Two

Appendix C: Outlet Z-Uno Firmware Code

```
/* WiSSP MQP Outlet Firmware Code
   Property of the WiSSP MQP Team
   Project Start: 08/22/2020
   Project End: 03/06/2020
   Designed for the Z-Uno
*/

/*Description: This script is written to drive the functions of the WiSSP outlet device. The system is designed to take input from two
  analog sources, the voltage level from the circuit and a current reading from the current sensor. The data is then sent to the hub device
  as a Multilevel Voltage Sensor, Multilevel Current Sensor, and a Binary Switch. The switch can then be toggled by the user via the companion
app in order
  turn the connected circuit on or off.
*/

//Pins
#define CntSensePin A1 //Analog Data from Current Sensor
#define VlinePin A2 // Analog Voltage input from the circuit
#define RefPin A3 //Analog Vcc for current calculations
#define CircuitPin PWM3 //Binary signal to turn circuit on and off
#define LEDPin 13 //LED Pin

//Global Variables
#define voltConv 3.3/4096 //conversion from ADC value to volts
#define refBias 2.106375 //scales the outlet's input from the voltage divider to calculate peak line current
#define voltBias 61.2451 //scales the outlet's input from the voltage divider to calculate peak line voltage
#define numOfLoops 200 //number of loops in the checking programs
#define SecondsDelay 30 //The delay of the entire system itteration (in seconds)
#define voltageDivChar 2.092923 //characterized value of the voltage divider in the current sensor circuit
#define sensativity 5.181647 //The amps per volt of the current sensor
#define ampFix .450 //fixes the noise problem from the internal circuitry (can be removed if shielding and insulation methods are improved)
#define voltFix .15 //fixes the noise problem from the internal circuitry (can be removed if shielding and insulation methods are improved)

int a = 0; //raw Current
int v = 0; //raw Voltage
float vcc = 0; // raw voltage reference
bool circuitStatus; //True is on, false is off
float ar = 0; //current value in amps
float vr = 0; //voltage value in volts
float oldar = 0; //for fixing negative ampage issues

//Set Z-Uno to produce a voltage sensor, current sensor, and binary switch

ZUNO_SETUP_CHANNELS(
 ZUNO_SENSOR_MULTILEVEL(ZUNO_SENSOR_MULTILEVEL_TYPE_VOLTAGE,
                SENSOR_MULTILEVEL_SCALE_VOLT,
                SENSOR_MULTILEVEL_SIZE_FOUR_BYTES,
                SENSOR_MULTILEVEL_PRECISION_TWO_DECIMALS,
                getterVoltage),
 ZUNO_SENSOR_MULTILEVEL(ZUNO_SENSOR_MULTILEVEL_TYPE_CURRENT,
                SENSOR_MULTILEVEL_SCALE_AMPERE,
                SENSOR_MULTILEVEL_SIZE_TWO_BYTES,
                SENSOR_MULTILEVEL_PRECISION_TWO_DECIMALS,
                getterCurrent),
 ZUNO_SWITCH_BINARY(getCircuitStatus, setCircuitStatus)
```

```
);

void setup() {
  // put your setup code here, to run once:
  pinMode(CircuitPin, OUTPUT);
  pinMode(LEDPin, OUTPUT);
  pinMode(CntSensePin, INPUT);
  pinMode(RefPin, INPUT);
  pinMode(VlinePin, INPUT);
  Serial.begin(9600);
  digitalWrite(CircuitPin, LOW); //Circuit is default on
  digitalWrite(LEDPin, HIGH); //Circuit is default on
  analogReference(DEFAULT); //reference Gnd to Vcc (actually goes to about 3.08V)
  analogReadResolution(12); //changes the read value to 12 bit resolution rather than the default 10 bit
  circuitStatus = true; //Circuit is default on
}

void loop() {
  // put your main code here, to run repeatedly:
  ar = oldar; //update the current value
  //Take Voltage Input
  if (circuitStatus) {
    convertVoltageSense(); //take the voltage reading
  }
  else { //if circuit is off
    vr = 0;
  }
  //Take Current Sensor Input
  if (circuitStatus) {
    convertCurrentSense(); //take the current reading
  }
  else { //if circuit is off
    ar = 0;
  }
  //Send Input to Hub
  zunoSendReport(1);
  zunoSendReport(2);
  delay(SecondsDelay * 1000); //delay circuit for <SecondsDelay> seconds
}

bool isNegative(float x){
  //Takes a float and returns 1 if negative
  if (x < 0){
    return 1;
  }
  else{
    return 0;
  }
}

void convertVoltageSense() {
  //Takes the analog voltage value and converts it to a value in volts
  v = 0; //reset variable
  int temp; //temp value for reading the pin
  for (long i = 0; i < numOfLoops; i++) {
    temp = analogRead(VlinePin); //take reading from the voltage pin
    if (temp > v) { //check to see if this sample is the highest value
```

```
      v = temp; //replace the current value with higher value
    }
  }
  vr = (v * voltConv - voltFix) *voltBias; //convert highest analog value
}

void convertCurrentSense() {
  //Takes the analog voltage value and converts it to a current value for export
  //runs 20 samples and finds the largest to compensate for the half rectified wave
  a = 0; //reset variable
  unsigned int temp = 0; //temp value for reading the pin
  int sum = 0; //value for averaging the voltage reference
  for (int i = 0; i < numOfLoops*5; i++) {
    temp = analogRead(CntSensePin); //take a reading from the current sensor pin
    if ((temp > a) && (temp < 2500)) { //check if the current value is the highest value
      a = temp; //replace the current value with the higher value
    }
  }
  for (int i = 0; i < 1000; i++){ // find the average of the reference to avoid noise issues
    sum = sum + analogRead(RefPin); //add 30 values from the reference pin
  }
  vcc = sum / 30; //divide to find the average ref value
  float calc = vcc * voltConv * refBias; //determine the ref value in volts
  ar = (((a * voltConv - ampFix)* voltageDivChar) - (.10123 * calc)) * sensativity; //convert the highest measured analog reading
  if (isNegative(ar)){
    ar = oldar;
  }
}

//Getter and Setter Functions for Z-Uno Transmission
//WARNING: DO NOT ADD CODE LOGIC TO THESE FUNCTIONS
int getterVoltage() {
  int temp = vr * 100; //grab the float's first two decimal places as an integer
  return temp;
}

int getterCurrent() {
  int temp = ar * 100; //grab the float's first two decimal places as an integer
  return temp;
}

byte getCircuitStatus() {
  return circuitStatus;
}

void setCircuitStatus(bool newStatus) {
  if (newStatus) { //turn the circuit on
    digitalWrite(CircuitPin, LOW); //circuit on is digital low
    digitalWrite(LEDPin, HIGH); //LED on for posterity
    circuitStatus = newStatus; //update to new status
  }
  else {
    digitalWrite(CircuitPin, HIGH); //circuit off is digital high
    digitalWrite(LEDPin, LOW); //LED off for posterity
    circuitStatus = newStatus; //update to new status
  }
```

Appendix D: Smart Tech Questionnaire used for Market Research

Electrical Engineering Capstone Smart Tech Questionnaire

We are conducting this survey as part of our MQP to design and create a smart outlet system. Our findings and results for this research will be published by Worcester Polytechnic Institute. We hope to gather consumer data in the smart outlet market to better design our product. This survey will take approximately five minutes.Your name or personal information will not be collected or distributed in any way. Please know this is entirely voluntary and you can stop, skip questions, or refuse to answer any question at any time. If you have any questions or concerns, please contact us at gr-wissp@wpi.edu.

1. Do you have smart home products in your place of residence?

   A smart home product is an appliance or device that can connect, share, and interact with its user and other devices. (ie. Amazon Echo, Google Nest, Smart TV, etc.)

   Mark only one oval.

   - ◯ Yes
   - ◯ No

2. Do you use a smart outlet in your place of residence?

   A smart outlet is an outlet that grants the user the ability to turn the outlet on and off remotely.

   Mark only one oval.

   - ◯ Yes
   - ◯ No, but I would be interested
   - ◯ No, and I am not interested

3. Rank the importance of the following smart outlet features.

A smart outlet is an outlet that grants the user the ability to turn the outlet on and off remotely.

Mark only one oval per row.

	1 - very unimportant	2 - slightly unimportant	3 - neutral	4 - slightly important	5 - very important
Ability to turn outlets on/off from anywhere you can access the internet	◯	◯	◯	◯	◯
Ability to turn outlets on/off based on user set schedule	◯	◯	◯	◯	◯
Ability to track power usage throughout the day	◯	◯	◯	◯	◯
Aesthetically pleasing design	◯	◯	◯	◯	◯
Alerts user to abnormalities in power usage	◯	◯	◯	◯	◯
Easy set up/use	◯	◯	◯	◯	◯
Security	◯	◯	◯	◯	◯

4. Do you have any major concerns with smart home products?

If yes, Please list them in the "Other" option.

Mark only one oval.

◯ No

◯ Other: _____

5. How concerned are you about power usage in your place of residence?

Mark only one oval.

	1	2	3	4	5	
Not at all concerned.	◯	◯	◯	◯	◯	Very concerned.

6. What is the most you would be willing to pay for a smart outlet system that includes the features you identified as important?

A smart outlet is an outlet that grants the user the ability to turn the outlet on and off remotely.

Mark only one oval.

- ◯ $20-39
- ◯ $40-$59
- ◯ $60-$79
- ◯ $80-$99
- ◯ $100+

7. What is your age?

Mark only one oval.

- ◯ Under 18
- ◯ 18-24
- ◯ 25-34
- ◯ 35-44
- ◯ 45-54
- ◯ 55-64
- ◯ 65+
- ◯ Prefer not to answer.

Appendix E: Wireless Communication Protocol Research

https://ieeexplore-ieee-org.ezproxy.wpi.edu/document/5174403

- System implemented uses wifi to connect with user, but Zigbee to connect locally (like our idea)
- A virtual home pre-processes all communications before they are realised on the real home automation system. All communications are checked for security and safety before being allowed to continue to their respective destinations.
    - Interesting concept?
- Zigbee is low power and simple which makes it good for local communication, and wifi is more complex and standard, making it good for communication to user
    - Again, just what we were thinking
- Zigbee Coordinator creates the Zigbee network
- Zigbee end nodes communicate and ask to join the network
- Zigbee communications were much faster than wifi

https://ieeexplore-ieee-org.ezproxy.wpi.edu/document/8757472

- Security should be implemented in the design phase
    - A personalized routine inherently will reveal a user's personal info
- Encryption is necessary
- Zigbee has more latency than wifi
    - Zigbee also requires additional technologies and routers, making the system expensive
    - Zigbee devices only work with other Zigbee devices from the same manufacturer

Indices	*ZigBee*	*Z-Wave*	*Wi-Fi*	*Bluetooth*
Power consumption	100 mw	1 mw	High	10 mw
Range	100 m	30 m	1000 m	10 m
Cost	Low	High	Medium	Very low
Scalability	6000	> 6000	32	20
Interoperability	Same manufacturer	Different manufacturers	Wi-Fi compatible devices	Bluetooth compatible devices

- What method you choose really depends on your customer. Are they new to the smart home industry or do they already have several devices?

https://ieeexplore-ieee-org.ezproxy.wpi.edu/document/7917995

- They did what we did, but very basically

https://ieeexplore-ieee-org.ezproxy.wpi.edu/document/8463281

- Z-wave

- Proprietary protocol
- Should be able to program with Z-wave without super understanding the protocol
- Master must store all info about the slave (slave description, slave name)

Cost Analysis:

All of these prices are the cheapest options I could find, meaning they are often just small IC's that must be connected. There are more system-based solutions out there that may be good for prototyping because they will be easier to learn

Wifi:

- End device ($8.51) https://www.digikey.com/product-detail/en/silicon-laboratories-inc/AMW037/1586-1035-ND/6097112&?gclid=Cj0KCQjwrMHsBRCIARIsAFgSeI3HieSHkVOTl3K2t-UOq4Tg-RWIdEFsHk1449t1kH7BEdJIVdqA4VwaAtWCEALw_wcB

Bluetooth:

- Transmitter: $35 https://www.adafruit.com/product/3055
- End device: $10 https://www.amazon.com/DSD-TECH-Bluetooth-iBeacon-Arduino/dp/B06WGZB2N4/ref=asc_df_B06WGZB2N4/?tag=hyprod-20&linkCode=df0&hvadid=309777534894&hvpos=1o3&hvnetw=g&hvrand=2336741542553692228&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9001847&hvtargid=aud-801657747996:pla-350315945458&psc=1&tag=&ref=&adgrpid=58425267301&hvpone=&hvptwo=&hvadid=309777534894&hvpos=1o3&hvnetw=g&hvrand=2336741542553692228&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9001847&hvtargid=aud-801657747996:pla-350315945458

Zigbee: https://electricalfundablog.com/zigbee-technology-architecture/

- Coordinator
- Router: https://www.digikey.com/product-detail/en/digi-international/X2-Z11-EM-A/602-1173-ND/2275403
- End Device

Z-Wave:

- Router: $10 https://www.semiconductorstore.com/cart/pc/viewPrd.asp?idproduct=99953&utm_source=GoogleShopping&utm_medium=cpc&utm_content=Silicon%20Labs&utm_campaign=Z-Wave%20serial%20interface%20module%20with%20antenna&gclid=Cj0KCQjwrMHsBRCIARIsAFgSeI1HOK83d36nMdzw_cap4Ukssn5gL7CijGoVmObFfxlumbyDtY9K5bUaAl6bEALw_wcB
- End device: $11 https://www.silabs.com/products/wireless/mesh-networking/z-wave/700-series-modules