Energy Guard

Home Energy Management and Security System

Senior Design I - Fall 2015 - Group 13

December 10th, 2015

Tyler Ensey
Gabriel Holland
Omar Mohammed
Spencer Sullivan

Table of Contents

1 Executive Summary

The purpose of this document is to detail the research and proposed design for the Energy Guard senior design project. To cover the semester long research conducted by the project's group members, the document will outline common industry-wide standards, summarize the systems and sub-systems of similar designs, conduct an in depth analysis of hardware options to be used to implement the design, provide plans and analysis of hardware and software designs to be implemented, and detail cyber security procedures and protocols.

The discussion on implementation will be divided into two sections, hardware and software design. The hardware design section will feature PCBs outlining the overall layout and use of physical components while discussing the functionality of each component, and its importance to the overall design. The section devoted to software development will cover, in depth, the sections code within various MCUs, the server, and the mobile/web application, detailing data flow, page flow, objects, table content and the functionality of each sub-section where applicable.

Energy Guard is a series of devices which can operate independently or cooperatively, collecting data to secure and/or limit the electrical power consumption of interior spaces while also providing clients with application-generated info-metrics of their total power usage. Composed of three types of devices, power strips, entry mounted sensors, and wall-mounted thermostats, together they allow for interior temperature control systems and electrical appliances to be powered or manipulated remotely from a client's internet-connected smart-phone. Device states can be altered either by a set schedule, by the number of occupants in a room, at the whim of the user, or any combination of the three.

With electrical appliances accounting for an increasingly high amount of energy usage due to the popularity of embedded systems, personal computing, and mobile phones, one of our project's devices devotes itself to directly reducing appliance energy consumption. The home power strip has the ability to turn on or off individual appliances by denying power to a specific electrical socket or even all sockets on the device. There are similar products on the market today, but they are not wirelessly controlled or do not have the ability to control the power flowing to individual sockets. Adding needed flexibility and adaptability to our product gives clients the choice of leaving some appliances on out of necessity, personal preference, or even for security, to deter unwanted guests, while still being able to control and monitor the use of their appliances while away.

The second device devoted to the project focus of energy consumption is the coupled thermostat and dryer monitor. Since heating and cooling interior spaces is the largest major contributor of commercial and residential energy consumption, it became apparent that temperature control systems also needed to be carefully regulated for our product to effectively create more energy-efficient spaces. By turning off HVAC systems when no one is in the building for an extended period of time or when the heat generating dryer is active, unnecessary power usage can be avoided by preventing the HVAC's unneeded or difficult attempts to establish a temperature equilibrium.

The initial purpose for including above entry mounted sensors was to count the number of people within a room for the sake of alerting other devices to either enter wake-state or hibernation state to conserve energy, and simplify the overall design, making appliance scheduling less complicated, and solely reliant on the above entry mounted sensor data. But when further analyzing the possible needs of building owners it became clear that these above door mounted sensors could serve a dual purpose as both a monitor for the related power systems and as a monitor for home security by distinguishing exterior entryway sensors from interior ones, and giving them increased responsibility when the home's secure mode is activated.

The final section of this project to integrate hardware with an easy-to-read user-friendly graphical-user-interface, is the web page and user application which will allow users to access their account through a secure, highly tested, login page or to create an account. From there, users can register devices to their account and buildings, track energy usage of specific appliances plugged into specific sockets, the usage of specific rooms, all the way up to the entire structure. As well as power-tracking, the user can regulate device schedules for both power usage, and security, manually activate and de-activate both devices and security modes, and add or remove users from their building group in they have a building administrator status.

1.1 Motivation

Today, the United States is facing many ever-encroaching energy-related problems. Warnings by climatologists about humanity's global impact on the environment have resulted in an expanding search for alternative and renewable methods to supply communities with power.

Shortly after the beginning of the new millennium, a rapid rise in the demand for sustainable energy sources and energy efficient appliances has transformed the market, but alternative energy today is neither cost-effective nor reliable enough to meet the world's energy needs. Although total household energy consumption has consistently declined in the United States, the introduction of personal electronics, embedded systems, and energy efficient, electrically powered, appliances are shifting the majority of the burden onto our electrical power grid.

With an already strained infrastructure and increasingly volatile weather patterns due to climate change, US residents will be at greater risk of experiencing power outages and blackouts in the future. In response to these looming and interconnected crises, when polled, seventy-nine percent of Americans described energy efficiency as a top priority. Although, in another poll, the number of Americans engaged in energy saving measures was shown to have decreased steadily over the last four years.

Americans aren't ignorant about the many methods of lowering their global footprint, and the popularity for alternative energies, especially solar and wind power, is very high. So why aren't more Americans trying to increase their home's energy efficiency? We believe this disparity between intention and action is caused by high costs and the hassle of installation.

Still, we cannot ignore the true cause of our strained infrastructure and changing climate. It isn't our inability to supply enough sustainable energy, but ultimately our ever increasing ability to consume it. Convincing others to reduce unnecessary power consumption should be our main priority in slowing climate change.

As a team, we plan to design an easily-installed home energy management and security system. By using a network of sensors and power strips, ordered and interconnected through an online database, users can log in, adjust settings, turn on and off the flow of power to individual devices, read current, daily, monthly, and annual statistics of their power usage, and activate and deactivate their security system, all from the convenience of their mobile phone.

The power strip will record and wirelessly relay power usage, be able to stop the flow of power to individual outlets both by user command and based off of doorway sensor transmissions, and provide surge protection to all physically connected appliances. If no one is in a room after a short period of time, all devices, unless specified by the user, plugged into the power strip will be turned off.

Above doorway sensors will serve two purposes. The first purpose is tracking the number of people present in a room and alerting all in-room power strips when that number drops to zero, or when it rises from zero to one. The second purpose is if someone has broken a window or been detected passing through an exterior doorway while the network is a 'secure' mode, a notification should be sent to the user's phone with an option to call the police. To accomplish these tasks, the use of motion sensors and glass breakage sensors must be implemented, and doorway sensors must be differentiated as either interior or exterior entryways.

All devices can be organized by home and broken up into a subsection of room networks, with the option of having personalized settings for each outlet, device or room network. Each outlet, device, and room network should be able to be remotely powered on or off by the user through the app, and energy statistics on these groupings should also be available.

As a group, we believe that if this product can be easily self-installed at a cost which is competitive or lower than other similar products, can accurately inform homeowners of their power usage, is light weight, and effectively alerts residents to home intruders, it will be able to meet the needs of American homeowners.

2 Design Requirements

Our design requires 3 devices and a server to communicate correctly and efficiently using an android app. The server will act as main data hub. All 3 devices in our system will be sending data over Wi-Fi to the server through a separate MCU for the device. **Figure 2-1** shows the block diagram for our power strip including the server and App. Each outlet in the system will be able read power for the device that is plugged in. Each outlet also has surge protection connected to protect from power surges form the power supply. The power reading will be sent to the server through the MCU. The database will house the long term data that the user can access through the android app.

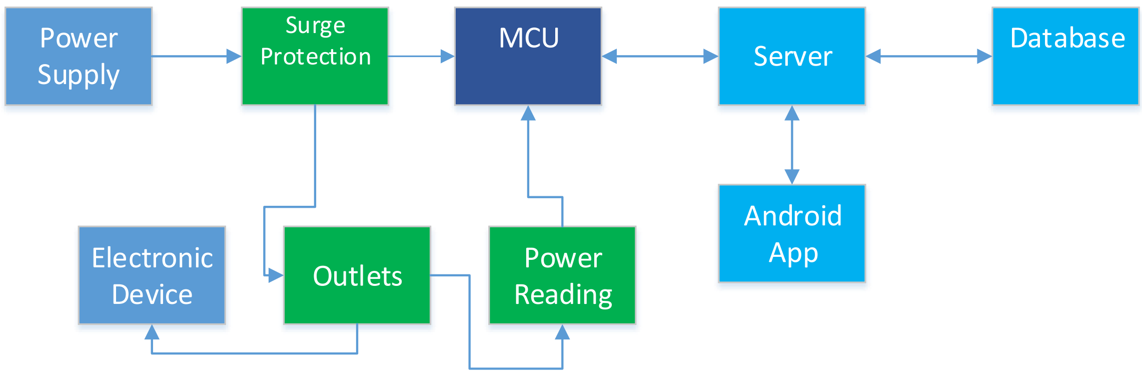


Figure 2-1: Power strip Block Diagram

Figure 2-2 shows the block diagram for the other two devices on our system. We will use as least 3 sensors connected to a MCU. These sensors will be used to determine if anyone is a room so the system can react to changing conditions. The HVAC and Dryer system will have two components to represent the function of a Dryer and HVAC thermostat. Due to design constraints, we could not afford to buy a dryer or create a HVAC system. Overall the server, the database, and the android app act as a hub for all 3 systems to send and receive data over Wi-Fi.

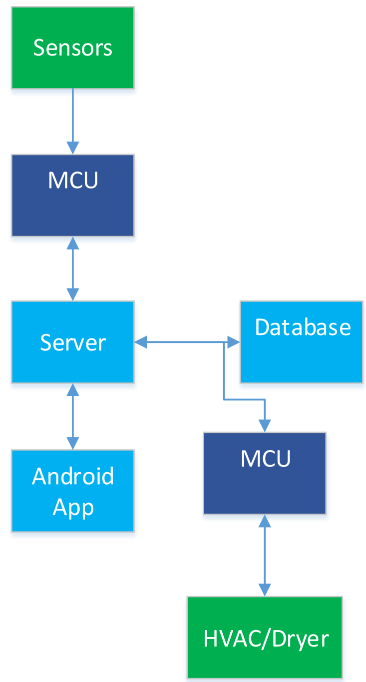


Figure 2-2: Mounted Device and HVAC/Dryer

2.1 Design Goals and Objectives

The proposed project is for a home energy management system. This system will be easy to implement in any home. The system will consist of 3 devices:

1. 3-outlet power strip
2. Door Mounted Sensors
3. HVAC and Dryer Control

The Power Strip will consist of:

- 3 or more outlets that will be able to read voltage, current and power from each outlet connected to the strip.
- Wi-Fi connectivity to a device running an Android operating System.

The 3 outlet device will consist of a Wi-Fi MCU which will send power measurements of the system to the database. The system will be able to provide power measurements of each connected appliance. Control of the system will be accomplished by the user through an Android phone connected to the Wi-Fi network. The phone will be able to turn off any outlet for the system effectively turning off the appliance. The app will also show power readings with a GUI so the user can know how much power is being consumed on each plug and overall power of all appliances.

The door mounted sensor will consist of:

- A separate wireless MCU
- 1-3 sensors to determine room occupancy

The sensors will be used to determine if anyone is still on the room. If no one is detected in the room then the devices connected to the power strip could be turned off to preserve power. This can also be used for security features also although that is not the focus of the design. This part of the system will be a door mounted device. The device will be separate from the main 3-outlet device mentioned earlier. The mounted device will have its own MCU which will control sensors monitoring the room.

The HVAC and dryer control will consist of:

- A thermostat to control temperature settings
- The thermostat will have Pentiometer or push button controls
- 4 Seven segment displays
- Two LED's to display cooling and heating
- A separate wireless MCU
- An LED to act as a dryer control

The purpose of this device is to control the two high power consumption devices HVAC and a dryer. The user will have the ability through the app to raise or lower the temperature of the HVAC so it will not run as often and save power. This will require a thermostat that is able to connect to Wi-Fi or the system itself. We would also like to monitor a dryer that would be connected to our system. For the purposes of our limited budget the dryer will be represented in our circuit by an LED which we will control through the app. The device should have some logic to tell the dryer to shut off if the AC is running or vice/versa. The HVAC system will be controlled by a thermostat. The thermostat will consist 4 seven segment displays to display a two digit number. Our temperature will increase or decrease either by a pentiometer wheel or push buttons. Two LEDs one blue and one red will light up when either the system is cooling or heating.

Figure 2.1-1 shows a detailed block diagram of what we want our system to be. For simplicity each of the 3 microcontrollers are combined into one box. Each Relay will be powered from the power supply which goes into greater detail later in this document. Also the surge protector will be in-between the power supply and relay components. Each outlet will be have power read separately and sent to the microcontroller reporting. The microcontroller can control each relay separately. This will offer a greater amount of control for the end user and the app.

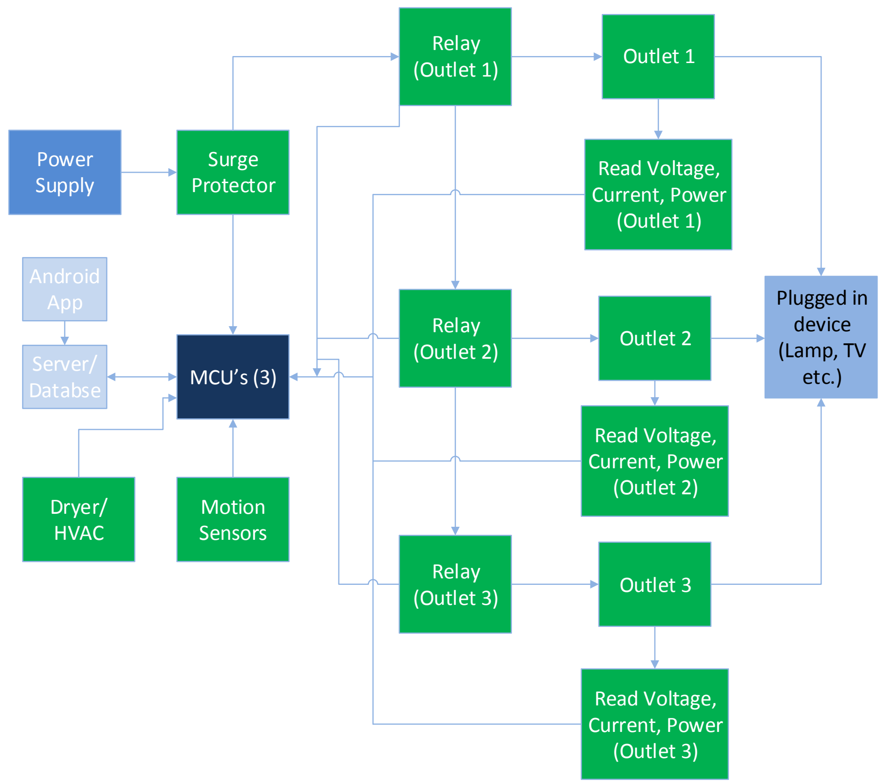


Figure 2.1-1: Detailed system

2.2 Design Constraints

This project has a number of design constraints. Namely we have economic constraints in the form of budget and pricing and the time to complete this project is very short. Also, this project must comply with a number of engineering standards. Standards will be discussed in the Standards section of this document. Our budget is tight at just under $400.00. We have 3 devices to buy parts for so we must choose parts that are cheap but will work as efficiently as possible for us. Time is a large constraint for our project. We must complete the implementation of the design in about 3 months. This fact causes us to be simple with design choices and pick parts that will be easy for us to implement into our system. Also our project needs to be affordable and able to install into any house. Anyone regardless of the age or set up of their home will be able to install and use this energy management system. So the constraint is that the system needs to be standalone and mostly plug and play. The end user should not have to break into any walls or change anything about the home wiring.

Design constraints also be Ethical and political. We originally had an idea of using cameras in our design to offer a security feature. The cameras would also communicate with the server and be able to send a live feed to the app. We ultimately didn't go with this feature because of security concerns. Offering cyber security for something as private as a home camera system has a large number of government regulations that we would have to observe. This would be too difficult to explore and implement with the other parts of our design and the limited amount of time we have to complete the project. Also there are ethical concerns with the camera system. Our app has the ability to be shared with others in household such as siblings or roommates. Ethics comes into play if there are cameras in rooms or private spaces that are able to be viewed on the shared app by other household members. We decided against cameras because of the mentioned reasons. Although, cyber security of data is still a constraint in our system and needs to be addressed appropriately.

A design constraint we will encounter are social constraints. This is a group project so finding a time to meet can be a challenge. We need to meet as a group often to go over the different challenges of the project. We are not able to meet as often as we would like due to conflicts. The reason for face to face meetings is to go over design challenges and make smart decisions on how to make our project a reality. Agreeing on all design choices is a constraint also. We all have differing opinions on how to move forward. In the end, we have to agree with the majority and make decisions that will help our project succeed.

3 Standards

Standards are important in just about every part of our life. They protect our health, our safety, our environment. They also have a profound effect on expanding markets, nationally, regionally, or internationally. They also help in reducing costs, time it takes products and services to reach the market, and they allow for competitive advantage. Standards, or more specifically engineering standards, are very important when it comes to developing a product. Therefore, it is obvious that for the development of our product we must follow certain standards. We followed various standards that pertained to Wi-Fi, communication protocols, electrical, and safety requirements.

3.1 IEEE 802.11

The IEEE 802.11 standard governs anything that is associated with wireless or Wi-Fi signals. The standard is named for the Institute of Electrical and Electronic Engineers (IEEE). The standard covers specification for operating wireless local area network (WLAN) traffic in the 2.4, 3.6, 5 and 60 GHz bands. The standard cover a myriad of situations in which they name protocols. The documentation names them with the nomenclature “802.11xx” where the xx is a letter or number combination. It is not feasible to cover all of the separate protocols in this document but I will mention the ones that may apply to our project.

At least in the United States, almost every router is tagged with the specification “802.11b/g.” This string is actually pointing toward two protocols for the IEEE 802.11 standard. The 802.11b protocol means that the router will adhere to the max data rate and the bandwidth assigned in the standard. This protocol states a max data rate of 11 Mbits/s and operates in the 2.4 GHz band. This router with the “b” protocol will have interference with anything in the same band including cordless phones and Bluetooth devices. The “b” protocol received an upgrade in the early 2000s to the “802.11g” protocol. This protocol has a higher raw data rate of 54 Mbits/s and still operates in the same 2.4 GHz band. One more protocol worth mentioning is the “802.11n” protocol. This protocol can not only operate in the 2.4 Ghz band but also the lesser used 5 GHz band. The 2.4 GHz is so widely used by a large number of wireless device that is has become crowded with traffic. The 5 Ghz range is not used as much but comes with a tradeoff. Using the 5 GHz range will decrease the wavelength of the modulated signal. This means that your wireless signal will not be able to penetrate as far as the 2.4 GHz band. The “n” protocol also operates at a max data rate range between 54 and 600 Mbits/s. The Wi-Fi solution we are using for our project is the embedded Wi-Fi in the CC3200 microcontroller. This microcontroller embedded Wi-Fi solution conforms to the 802.11b/g/n protocols. There is no need to change or add any extra hardware to conform to the standard.

3.2 MQTT Protocol

MQTT stands for Message Queuing Telemetry Transport. This protocol operates on the TCP/IP protocol. MQTT is a publisher-subscriber messaging protocol that is can operate in unideal communications environments. Such environments include low bandwidth, unreliable networks, or low-latency among others. The purpose in design of the MQTT protocol is to offer reliability of sending data in an otherwise hostile environment. Our system most likely will not suffer low connections speeds or bandwidth issues but what our system will provide a mobile application to interface with our devices. MQTT is prime to implement for mobile apps since it has low power consumption and bandwidth for a mobile signal is not always readily available. The standard for MQTT is held by a standards organization called “OASIS.” The current published MQTT standard is version 3.1.1 published in October of 2014.

3.3 Electrical (Standards)

There are a large number of Electrical standards enforced today. The electrical standards that apply to our project are IEC 60320, and Socket Types. IEC 60320 outlines the specifications for connectors. Each connector must meet IEC max ratings in order to be IEC compliant. Also the Underwriters Laboratory (UL) organization also will certify electrical components to be compliant with safety. Any socket connector we use should have an IEC and UL compliant stamp or it may be unsafe. According to UL and IEC connectors should have max voltage of 250V and be able to handle about 16 A of current. Of course this for connectors used in the US. Socket Types have also been standardized by Region. For the US we must use Type A and/or Type B sockets and supply at least 120 V at 60 Hz AC voltage. More detailed information is covered later in this document.

3.4 Safety

Having and following safety standards are an integral part of ensuring the safety of our product to the user. There are many safety standards for a variety of different products, activities, or processes. Even more specifically, there are abundant amounts of safety standards just for electrical products. Our product followed the following sections of the standards that pertained to our product:

a. UL3111-1
b. IEC1010-1
c. OSHA1910 §§ 303 – 305

The UL3111-1 standard encompasses safety standards for electrical devices that are used within professional, industrial, and/or educational environments. These safety standards are used for devices that do the following:

- Measurement and testing
- Control
- Laboratory use
- Accessories intended for use with the above

The IEC1010-1 standard is described as safety requirements for electrical equipment for measurement, control and laboratory use. IEC1010-1 states that the voltages 30 V_{RMS} or 60 V_{DC} are dangerous. Design constraints that take into flammability and heat are also taken into account.

The Department of Labor's (DOL) Occupational Safety and Health Administration (OSHA) created Occupational Safety and Health Standards known as OSHA1910. For safety, we are mainly concerned with sections 303 – 305. Some main parts of each section that pertained to the safety standards of our product included:

- § 1910.303(b) – Examination, installation and use of equipment
    1. Examination
    2. Installation and use
    3. Insulation integrity
    4. Interrupting rating
    5. Circuit impedance and other characteristics
    6. Deteriorating agents
    7. Mechanical execution of work
    8. Mounting and cooling of equipment
- § 1910.303(c) – Electrical Connections
    3. Splices
- § 1910.303(d) – Safety of arcing parts
- § 1910.303(f) – Disconnecting means and circuits
    1. Motors and appliances
    2. Services, feeders, and branch circuits
    3. Durability of markings
    4. Capable of accepting a lock
    5. Marking for series combination ratings
- § 1910.304 – Grounding of various devices
- § 1910.305(g) – Flexible cords and cables
    1. Use of flexible cords and cables
    2. Identification, splices, and terminations

The devices that we buy should follow these various safety standards. Devices that follow the UL3111-1 standard will have a UL label, and devices that follow the IEC1010-1 standard will have an IEC label.

3.5 PCB Design

We will have to have 3 printed circuit boards (PCBs) made for each of our devices. Manufacturing a PCB can be costly depending on which manufacturer is chosen to make the board. Also a problem with PCB creation is the time it takes to create and ship the board. Some manufacturers have a lead time of a month or greater. This is too long for our purposes. An ideal lead time would be 3-5 days so that we can properly test the board and reprint if needed. A good choice for us is the company Advanced Circuits. The company has a ship time of 5 days. So that is perfect for our project.

Advanced Circuits caters to students and offers a number of services to help students succeed. The company offers a large discount to students making PCBs. They offer $33.00 to make a 2 layer board and $66.00 to make a 4 layer board. Our budget is really tight so we will try to make our project work with 2 layer boards since they are cheaper to make. Advanced Circuits also offers a free PCB layout software tool. This is a possibility for us to make our design files with. The website also accepts Eagle design files also. We may use Eagle since a large amount of documentation exists on how to use this program. Finally, Advanced Circuits offers a free PCB design check called “FreeDFM.” The tool will let us upload our PCB design and it will check the design for any issues that may cause delays in the manufacturing process. FreeDFM will also make sure we have all necessary files the company needs to print out boards. Overall, Advanced Circuits is a good choice for us to economically print our boards and give us the most opportunity to succeed.

3.5.1 PCB Population

Printing the board is only half of the battle. The board now has to be populated with all of our components. This can be very costly depending on the volume of components needed on the board. Advanced Circuits also provides PCB assembly services. They do not give any prices on how much the service costs. We will have to contact them for a quote once we have design files. Advanced Circuits have a number of points to consider for PCB assembly.

- The PCB should have at least 100 mm border to around it.
- Attempt to space out the components evenly along a horizontal and vertical path. If possible, the same or similar components should face the same direction. Make sure to place components at 0 or 90 degrees.
- Consider the volume that needs to be run. Not only the number of PCBs but also the number of components on the PCB. It will be more cost effective to manually populate a low volume PCB. Automated assembly is needed for high volume boards.
- Consult with the PCB assembler to insure the finish will work with the assembly process

- Attempt to minimize trace lengths wherever possible.

Given these points, we may only need to have the PCB for the power strip sent in for automated Assembly. The other two devices have such a low volume of components that it will be more cost effective to manually assemble the board.

3.6 PCB Design Comparison

While it seems Advanced Circuits offers all of the services we need, it is wise to explore other options. Another company ExpressPCB is a good choice to consider. ExpressPCB offers a price of $51.00 to have 3 boards created and shipped. There are limitations to this price. ExpressPCB names this their “MiniBoard” service. The terms of the MiniBoard Service are:

- 3 identical boards will be shipped
- All 3 boards are 2 layer
- Max board size is 3.8 x 2.5 inches
- No solder Mask

The price and lead time are the best about this option. If the order is placed by 2:00 PM, ExpressPCB ships the 3 boards the same day. The tradeoff for this low price is a fixed size board. We would need to try and fit all components on this small board size. This may not be feasible for parts of our design. Also something to consider is no solder mask. A solder mask layer is used to help reduce or eliminate what are called solder bridges. A solder bridge is an unintended connection between two conductors on the PCB. Having no solder mask could be risky for our design since we are all novices at soldering. It is helpful to know that we have 3 boards to get the soldering right. Just like Advanced Circuits, ExpressPCB also offers free PCB software. The software will also give a direct quote once the design is complete. It is unclear if the software will check our design for manufacturing errors like Advanced Circuits. Although Advanced Circuits offers a higher price, the boards will be better quality and will give us a better chance of achieving our project design goals.

3.6.1 PCB Assembly Comparison

Our first choice for PCB assembly is Advanced Circuits as mentioned earlier. The company will match any competitor pricing. The other option is manually soldering the board ourselves. This is only feasible if the volume on components is low enough so we don’t spend the entire time of the project trying to solder our boards. Mentioned earlier, ExpressPCB does not offer solder masks. This would be a problem with board assembly. Advanced Circuits requires that a solder mask be in place for the reflow process. No solder mask means they will not be able to fulfill the order. Any board bought from ExpressPCB will either need to have a solder mask put in place or we manually solder the board on our own. Both of these options are not ideal since they may cost us both money and time. Also we have heard that the TI Innovation lab in engineering 1 at UCF will soon be able to use a reflow process to populate PCB Boards. Any attempts to confirm that this will be available for our project milestones have been unfruitful. We will continue to find out more information as this would be a very useful way for us to populate our boards.

4 Influential Designs

This section discusses other system designs that played a role in the design of our system. We did not “copy and paste” these designs but each device mentioned helped us to know what concepts we could use and what worked for their system.

4.1 TI Smart Power Strip Design

During our research, we found a Three Outlet TI Smart Power Strip. This design is very similar to what we want to achieve as our end result. **Figure 4.1-1** shows a high level block Diagram for the TI Power strip. The basic design of this power strip is exactly what we need to accomplish our project goals. The power supply will come from an AC socket from a domestic wall outlet. The power supply will be discussed in greater detail later. Each of the 3 input sockets have a relay that controls power to each individual socket. The relays are controlled by a relay driver shown in the block diagram. The communication block is important since access to the MCU is critical in the functionality. This design uses the MSP430i2040. The MCU comparison for our project is found later in this document.

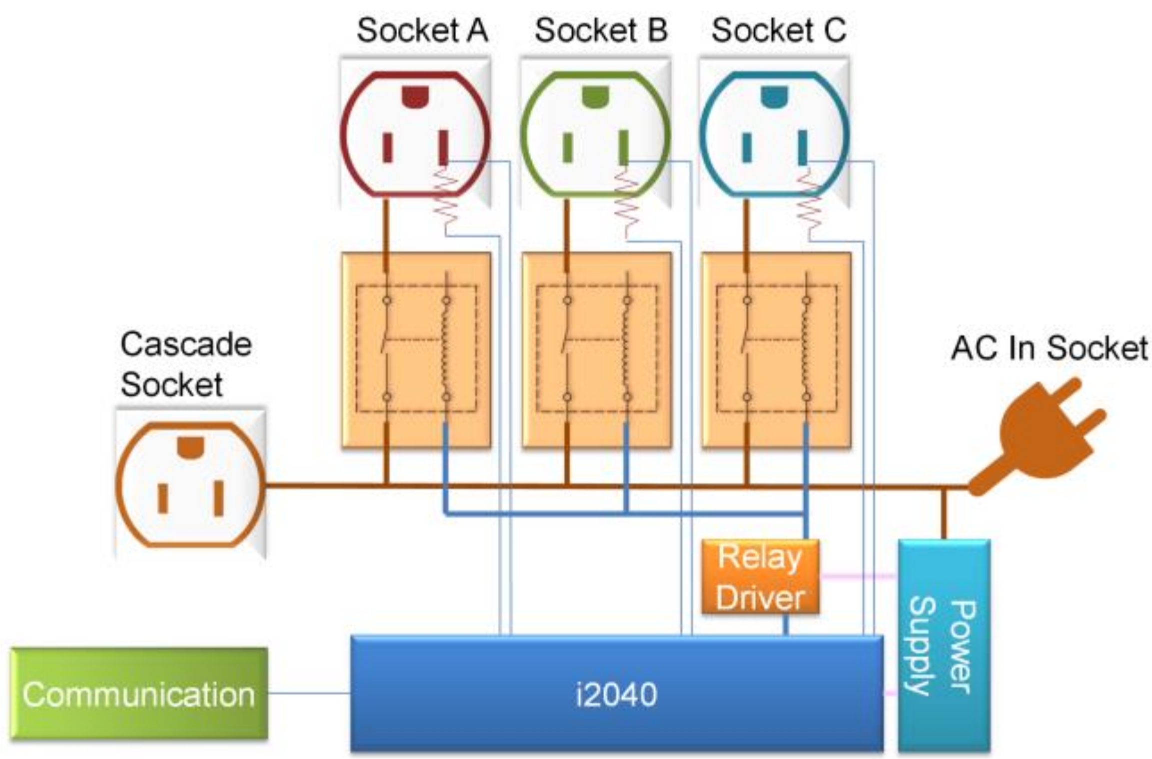


Figure 4.1-1: TI Design Block Diagram
Reprinted with permission from Texas Instruments

4.1.1 TI Design Choices

The strip uses relays to switch power on and off to the circuits. The relay driver will control when to turn off or on each outlet or component in the design. The ULN2003LVDR 7-channel relay driver is what TI uses for their design. This driver is available to a large range of microcontrollers due to its support of CMOS logic using a range of 3.3V to 5V for switching applications. The strip monitors power with four different circuits. There are three identical circuits to measure current in each of the 3 sockets, and one circuit to measure voltage. **Figure 4.1.1-1** shows the 4 circuits used. Inside the red box are shunt resistors for the current measurements. These shunts will be used as ammeter shunts to measure the current. The shunt is in series with the Voltage and current coming into the circuit. A property of a shunt is that voltage drop across the resistor is proportional to the current flowing into the load. For example, if a current shunt is rated at 50 A, 50 mV then resistance is known for the shunt as 1 ohm. Since the resistance is known a voltmeter can be used to display the voltage across the resistor and using Ohms law calculate the current. The two yellow boxes show low pass filters in each of 3 current measurement circuits. The two filters attenuate radio frequencies that interfere with the analog to digital converters for the MCU. The black box shows another filter that has a bandwidth of about 10 kHz. This is the filter for the band of interest in the circuit.

The light blue box in **Figure 4.1.1-1** shows a resistor ladder network for the voltage measurement. This is a voltage divider between the 3-330k resistors and the 1.5k resistor. The purpose of this divider is to prevent arcing between the terminals. This effectively keeps the voltage low enough so the MCU will not be damaged by a high voltage. The diodes D1(A,B,C) and D8 all offer protection in the circuit. These diodes are optional but are a good idea to keep so there is no danger to the MCU becoming damaged. The D8 diode is more important than the other 3. This diode is more likely to see higher voltage spikes than the other diodes.

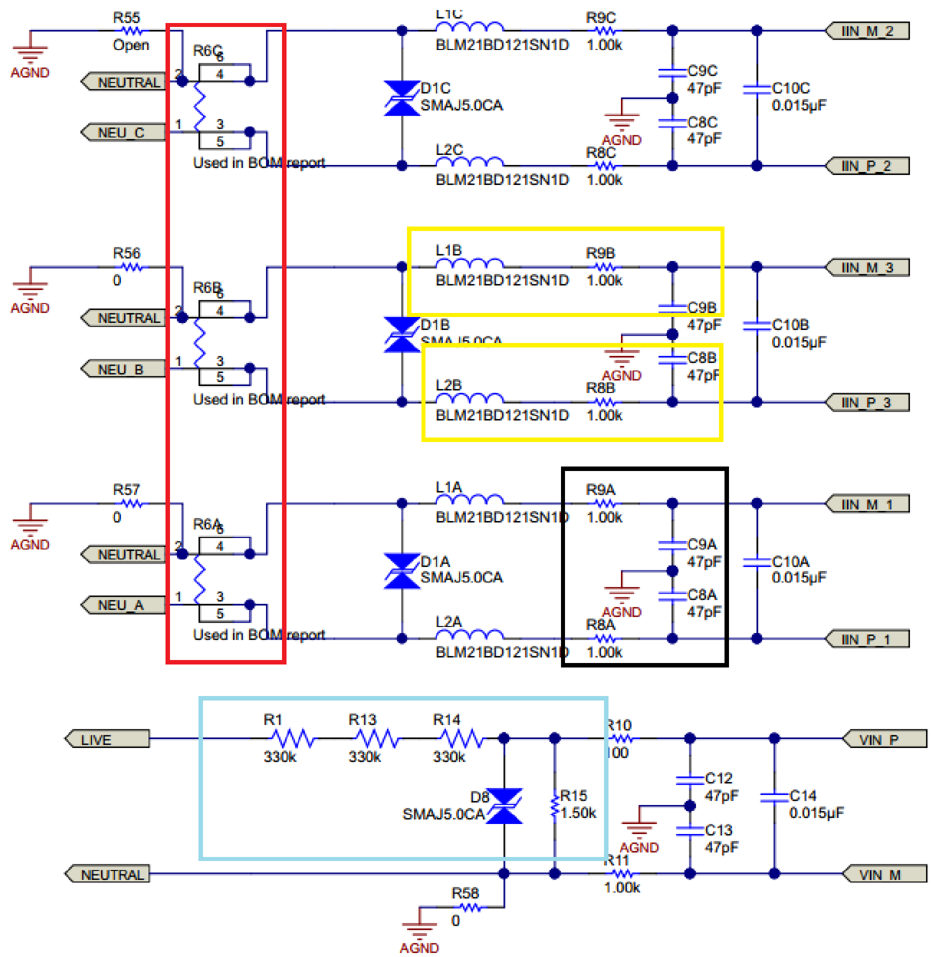


Figure 4.1.1-1: Current and Voltage Circuits Reprinted with permission from Texas Instruments

Figure 4.1.1-2 shows how a shunt resistor measures current while in series with the voltage source. These shunt resistors are sensitive to temperatures. If the temperature of the component increases too high, then the accuracy of the measurement will be off by a significant amount. Keeping the resistance of the shunt low will help to keep the component from overheating. Also the resistance should be low to increase accuracy of the measurement. The shunt should be in the range of <.1m Ω. This needs to be true so the voltage measured can be used to calculate the current using ohms law. A higher resistance will cause the calculation to be incorrect.

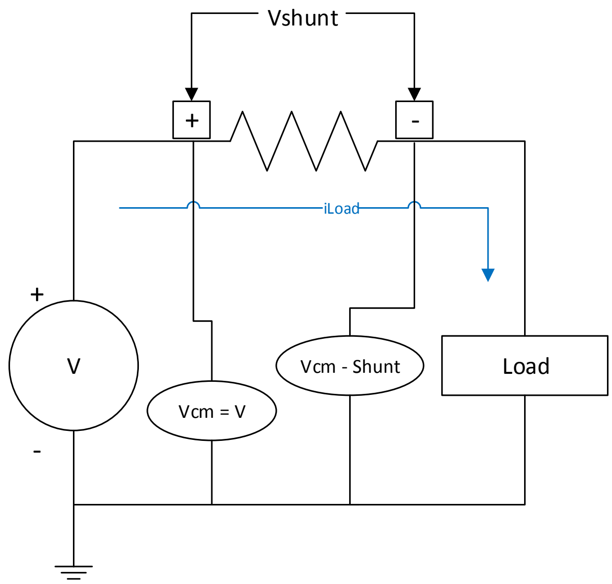


Figure 4.1.1-2: Shunt Resistor

Figures 4.1.1-3 (a)-(c) shows a small simulation on how the shunt resistor measures the current. Using a small resistance will show a voltage close to the source. The source used in the simulation is 120V at 60 Hz which is what our power supply will be. The AC current measured with the DMM is 12µA. The Voltage across the shunt is 120 V. So using Ohms law $\frac{12\mu V}{.1m\Omega} = 120mA$. As shown in **Figures 4.1.1-3 (a) and (b)**, the correct current is in face 120 mA. The circuit in **Figure 4.1.1-3 (c)** shows a 5 ohm resistor. This circuit measures 596 mV which calculates to a current of 119.5 mA. Although this current is still close to the correct value, you can see the relationship that the shunt resistor should be lower values to maintain the correct current.

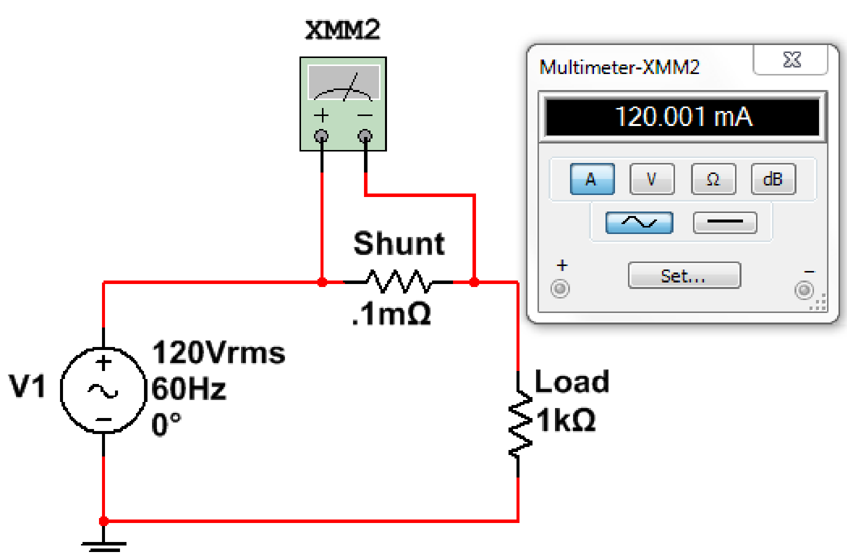


Figure 4.1.1-3 (a) Current through Circuit with small resistance

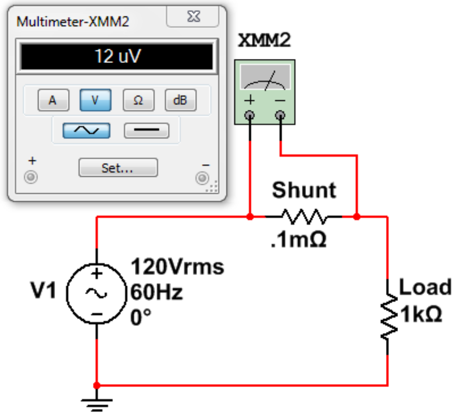


Figure 4.1.1-3 (b) Voltage through Circuit with small resistance

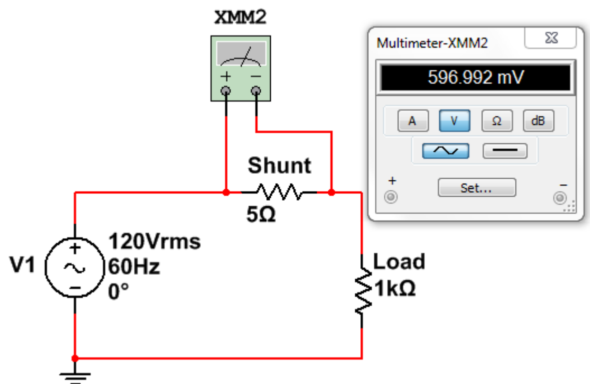


Figure 4.1.1-3 (c) Current of through Circuit with larger resistance

4.1.1.2 Expanding on the Design

Although this is a great starting point, this power strip does not do everything we want. This strip has no functionality for Wi-Fi or Bluetooth. Because of this fact, we have to modify the design so that we can add the sensors and inputs that will allow remote access of the system and data collection. Also the design uses all of the open inputs and outputs for the microcontroller. So we need to use a different microcontroller or find a way to increase the inputs or outputs to the microcontroller using other means.

TI's design shows a cascade socket. From what the block diagram shows, this looks like a socket that will always receive power from the AC wall socket. So any device plugged into this socket will work but will not be monitored. This is not in line with our project goals so we will most likely not use the cascade socket. We would like every socket on the strip to be able to turn off or on and give power measurement readings to the end user. The communication block shown in the block diagram of **Figure 4.1-1** is able to access the microcontroller code via a ribbon cable. This will be very nice to keep and is not very costly to implement. Although, we will have to change the circuit a little from what TI shows since our microcontroller is different from their design. The TI design also uses a RS232 port for serial access. They use it to evaluate the system and change any current or voltage setpoints. This feature does not fall into the scope of our project.

Overall, the design of this project is just a power strip. Our design integrates two other devices to communicate together and create a complete system. Also a key part of our design is the App that will be able to control parts of our system. This part of our project is not implemented by the TI Design. Expanding on this design will give us the most opportunity to learn and help bring better home energy management then the original TI power strip.

4.2 Nest Learning Thermostat

The Nest Learning Thermostat, shown in **Figure 4.2-1** and referred from here on as simply Nest, was very influential in our design of our smart thermostat. The Nest includes many advanced features, including but not limited to the following: learning your schedule and what temperatures you like at specific times and days of the week, and automatically setting itself during those times; having built-in sensors that have a 150º field of vision to recognize when nobody is home and turning off the air conditioning (AC); these sensors also are used to recognize when you're nearby and gives you the time of day, literally; Wi-Fi compatible, so that you can set the temperature from your computer, smartphone, or tablet; and, a notification when you choose a temperature that is energy efficient. The functions that we will use include the sensors that recognize home activity and Wi-Fi compatibility, for thermostat control from an android phone. We will also use the aesthetic design of the Nest as a starting point for our thermostat design.

Figure 0-1.2-1: The Nest Learning Thermostat
Reprinted pending permission from Nest Labs

Nest Labs, the company behind the Nest, was founded in 2010 by Tony Fadell and Matt Rogers, former Apple Engineers. Fadell came up with the idea when looking for a new thermostat and realizing that none were what he wanted and that he could make desirable enhancements to the modern thermostat. Nest became a booming success, and in 2014 moved to farther reaches than just the United States, including Belgium, France, Ireland, and the Netherlands. On January 14, 2014, Google acquired Nest Labs for $3.2 billion, showing just how successful this company became and still is. This is a classic case of identifying a problem that affects a large amount of people and making a marketable product. Seeing their success, it was not hard to choose them as the smart thermostat that we would like to base our smart thermostat.

4.2.1 Hardware

The design of the Nest includes an LCD screen for readings by the user and the ability to be rotated left or right to change temperature colder (left) or hotter (right), and push functionality to set to cooling or heating. We took after the Nest in its design, the main difference is that we will not be using a LCD screen, instead we will have two sets of LED lights, one set that displays the actual temperature inside the house and another that displays the desired temperature, and pushing the display will light either a red or blue LED to represent heating (red) or cooling (blue).

The microcontroller unit used by the Nest is ST Microelectronics STM32L151VB ultra-low-power 32-bit MCU ARM®-based Cortex®-M3 MCU. This microcontroller has many nice features including the following:

- Ultra-low-power platform
  - 1.65 V to 3.6 V power supply
  - 9 µA low-power run mode
  - 214 µA/MHz run mode
- Core: ARM® Cortex®-M3 32-bit CPU
  - From 32 kHz up to 32 MHz max
- Reset and supply management
- Multiple clock sources including 1 to 24 MHz crystal oscillator
- Pre-programmed bootloader
- Development support
- Up to 83 fast I/Os (73 I/Os 5V tolerant)
- Memories
  - Up to 128 KB Flash with ECC
  - Up to 16 KB RAM
  - Up to 4 KB of true EEPROM with ECC
  - 80 Byte backup register
- LCD Driver for up to 8x40 segments
- Rich analog peripherals
- DMA controller 7x channels
- 8x peripherals communication interface
  - 1x USB 2.0
  - 3x USART
  - 2x SPI 16 Mbits/s
  - 2x I2C
- 10x timers
- Up to 20 capacitive sensing channels supporting touchkey, linear and rotary touch sensors

- CRC calculation unit, 96-bit unique ID

This is a great microcontroller for what we are trying to achieve, but we will compare a few different microcontrollers later in this document, and decide what will be used.

A Sensirion SHT20 humidity and temperature sensor was used for the detection of current temperature and humidity within the home. The SHT20 has low power consumption, with the typical power consumption being 0.5 µW in sleep mode and 0.9 mW when performing measurements. The SHT20 sensor detects humidity through a capacitive humidity sensor, and temperature is detected through a band gap temperature sensor. This reading of temperature and humidity is sent through a specialized analog and digital integrated circuit and outputs a digital signal in I^2C format.

The Nest also totes a Texas Instruments SN74LV4051A 8-channel CMOS analog multiplexer and demultiplexer. This multiplexer and demultiplexer is designed to run at 2 to 5.5 V. It also handles both analog and digital signals, with each channel allowing signals with up to 5.5 V (peak) amplitude. The various applications of this chip include signal gating, chopping, modulation or demodulation, and signal multiplexing for analog-to-digital and digital-to-analog conversion systems. The primary function of this chip in the Nest is for signal multiplexing for analog-to-digital conversion.

A lithium-ion rechargeable battery is used to power the system when the low-power connection from your home is not enough. Meaning that power is still maintained even when there is a power outage. The battery is rated at 3.7 V, 2.1 Wh.

The Nest has Wi-Fi compatibility, and is based on IEEE 802.15.4 and Wi-Fi 802.11 b/g/n. The Nest also contains near-field, far-field and ambient light sensors. These are used to detect activity within the home, to help with scheduling and auto-away function. This makes sure that the HVAC system is not turned off when the house is inhabited. And makes sure energy is not wasted while nobody is home.

4.2.2 Software

The Nest operating system is based on Linux 2.6.37. This operating system allows for the user to interact with the system by spinning and clicking the control wheel. One of the more advanced features is the ability to interact with the thermostat without any physical contact. The thermostat connects directly to the Internet, this allows for user control, while also allowing Nest Labs to apply updates that improve the overall functionality of the system.

5 Power Strip

As mentioned in the design requirements, our system will have three outlet power strip. This section will include the design specifics and also how the device will communicate with our system. We have shown how the TI power strip designed their power monitoring and what microcontroller they used. We have chosen a similar design for systems such as the relays and relay drive, power monitoring and LED design. What we need to change in the design is the microcontroller. TI's design offered no Wi-Fi so we have chosen a wireless microcontroller to solve the problem of data communication with other systems.

5.1 Wireless MCU

The TI power strip does not offer any wireless communication. Our project requires wireless LAN communication. We also need a new MCU with a larger number of outputs. A MCU with an embedded wireless seems like a perfect fit to meet our project requirements.

The CC3200 MCU is a single chip wireless option. It has a wireless network processor with an ARM Cortex processor. This solution meets all required standards by the 802.11 Wi-Fi standard. This MCU should be a very simple plug and play solution for Wi-Fi. To meet our project requirements we need more outputs than the MCU used for the TI power strip design. The wireless MCU has 27 General purpose Input/Outputs (GPIO). This is more than enough to fit our project goals. The MCU used for the TI strip is a mixed signal MCU. Meaning some of the I/O's are analog and these inputs are what was used for the Voltage and Current measurements of the circuit. The CC3200 MCU is not mixed signal and only lists I/O's as GPIO. Each input expects a digital number. The CC3200 has 4 analog-to-digital converters on board and we can use those to make any conversion necessary for out project.

Mentioned earlier, this MCU offers a near plug and play experience for wifi applications. The Wi-Fi MCU supports certain internet protocols namely mDNS, DNS, SSL/TLS + web server (HTTP). The Domain Name System (DNS) protocol makes sure any devices connected to a system will have an IP address assigned. DNS protocols have authoritative name servers to keep track of domain names so they can be mapped to IP addresses. We will most likely not have to use this function since the home router will perform any name server assignments and DNS functions.

With the data we are planning to send over a network, we need a way to secure the data. Using the Secure Socket Layer/Transport Layer Security (SSL/TLS) protocols will allow us to provide data transfer over the network securely. The data will be secured through encryption algorithms. Not only data encryption but the protocol offers server authentication also. So transfer between 2 servers would have to authenticate the session with some keys so that data can transfer securely. The MCU comes with an embedded hardware crypto engine for quick SSL/TLS implementation. This means that the MCU should be able to make a secure connection in less than 200ms. The RFC 5246 standard outlines the TLS protocol and how it is used and implemented.

The HTTP Web server protocol is a popular protocol used for applications that need access to the internet. Using this protocol there would be a Client browser and HTTP server. The Client browser will initiate a URL which will be mapped by HTTP to the server. The server in turn will map that URL to a document file or Application and send it back formatted to the client browser to be displayed. The RFC 2616 standard outlines the HTTP protocol and how it is used and implemented.

The CC3200 also offers advanced power management functions. Our design needs to consume as less power as possible so the usefulness of home energy management is not lost in powering the device. The datasheet lists two sub systems that are configured for power modes. The first is a Cortex-M4 application processor subsystem, and the second is a Networking Subsystem. The Cortex processor runs user programs from a serial flash. **Table 5.1-1** shows what power modes are available from the highest power mode to the lowest power mode. Our device will most likely be active most of the time and have little time to sleep due to the devices connected will be on most of the time. If all devices plugged into the main hub are off then the MCU can go into a sleep mode and wake if any GPIO ports become active. For example the user trying to turn on a device using the app.

Table 5.1-1: User Program Modes
Reprinted with Permission from Texas Instruments

Application Processor (MCU) Mode	Description
MCU active mode	MCU executing code at 80-MHz state rate
MCU sleep mode	The MCU clocks are gated off in sleep mode and the entire state of the device is retained. Sleep mode offers instant wakeup. The MCU can be configured to wake up by an internal fast timer or by activity from any GPIO line or peripheral.
MCU LPDS mode	State information is lost and only certain MCU-specific register configurations are retained. The MCU can wake up from external events or by using an internal timer. (The wake-up time is less than 3 ms.) Certain parts of memory can be retained while the MCU is in LPDS mode. The amount of memory retained is configurable. Users can choose to preserve code and the MCU-specific setting. The MCU can be configured to wake up using the RTC timer or by an external event on specific GPIOs defined in Table 3-1 as the wake-up source.
MCU hibernate mode	The lowest power mode in which all digital logic is power-gated. Only a small section of the logic directly powered by the input supply is retained. The real-time clock (RTC) clock keeps running and the MCU supports wakeup from an external event or from an RTC timer expiry. Wake-up time is longer than LPDS mode at about 15 ms plus the time to load the application from serial flash, which varies according to code size. In this mode, the MCU can be configured to wake up using the RTC timer or external event on a GPIO (GPIO0–GPIO6).

Table 5.1-2 shows modes for the network processing subsystem. This system is only active when packets have to be sent or received to transfer data. This network processor will sleep when no there is no network activity. When the MCU hibernates, the power consumption will be about 4 μA of current so power consumption will be small. A low power deep sleep (LPDS) mode will commune about 250 μA of current. When the MCU active and with network traffic the current consumption can range from 59 - 250mA. This MCU will help to keep our power consumption low for the applications we need to implement.

Table 5.1-2: Networking Subsystem modes
Reprinted with Permission from Texas Instruments

Network Processor Mode	Description
Network active mode processing layer 3, 2, and 1	Transmitting or receiving IP protocol packets
Network active mode (processing layer 2 and 1)	Transmitting or receiving MAC management frames; IP processing not required.
Network active listen mode	Special power optimized active mode for receiving beacon frames (no other frames supported)
Network connected Idle	A composite mode that implements 802.11 infrastructure power save operation. The CC3200R network processor automatically goes into LPDS mode between beacons and then wakes to active listen mode to receive a beacon and determine if there is pending traffic at the access point. If not, the network processor returns to LPDS mode and the cycle repeats.
Network LPDS mode	Low-power state between beacons in which the state is retained by the network processor, allowing for a rapid wake up.
Network disabled	

Another Internet protocol we need to comply with is the TCP/IP protocol. The Transmission control Protocol (TCP) take a message and forms them into smaller packets. The TCP layer of the internet protocol then re assembles these packets. The packets know where to go because to the lower internet protocol (IP) layer. This layer will assigns and keeps track of these addresses. Without this layer would be like putting an envelope in the mail with no address on it. TCP packets must be sent with a high amount of reliability. This means that TCP will not allow packets to just become simply lost. Any lost packets will be retransmitted until the packet becomes reliable. Each packet must be sent in order and reliably. Such connection that use TCP are applications such as online banking. 100% packet reliability is needed here since the data is very sensitive.

This is contrast with User Datagram Protocol (UDP). UDP operates the same way as TCP. The message is split into smaller chunks of data called packets and sent through the IP layer to its destination. The difference here is reliably of the packets being received. The receiving of packets does not have to be 100% reliable. This means that when a packet is lost it is not retransmitted. It is actually desirable to have this packet loss. The type of applications that use UDP are voice over IP or video streaming. In these applications any lost packet needs to stay lost and not come up later in the stream. The microcontroller we chose complies with TCP/IP and UDP protocol stacks. The microcontroller comes with 8 TCP or UDP ports. The datasheet mentions that the throughput for UDP connections are 16 Mbps and 13 Mbps for TCP. The data we will be sending to the app will be sensitive in the way that the user may make financial decsions according to the data received. So it is beneficial that we use TCP ports for reliability even though we are trading off data throughput.

5.2 Surge Protection

Surge protection technology is fairly simple. The excess voltage that is put through a circuit needs to be drained off through a ground wire to protect the devices connected to the outlet. Components are needed within the circuit on the “hot” wire to know whenever a surge happens and to drain to the ground. One application to consider is the use of Metal Oxide Varistors (MOV’s). MOV’s are cheap and fairly easy to implement in any circuit. A MOV will have a high resistance at low voltages and low resistance at high voltages. The semiconductors will be set up in the circuit so that at high voltages all of the current will flow to the ground wire. Under normal operations of lower voltage, the current will not flow to ground due to the high resistance. Figure 5.2-1 shows how an MOV network would be implemented inside of a power strip. The live or hot wire will be connected between the device outlet and the MOV network. The other side of the MOV network will be the power supply from the domestic wall outlet. Under normal operation, the MOV network will be passive and the power will flow to the outlet. If a surge comes in from the power supply, then the MOV network becomes active and drains any excess power through the ground wire effectively protecting the device plugged into the strip.

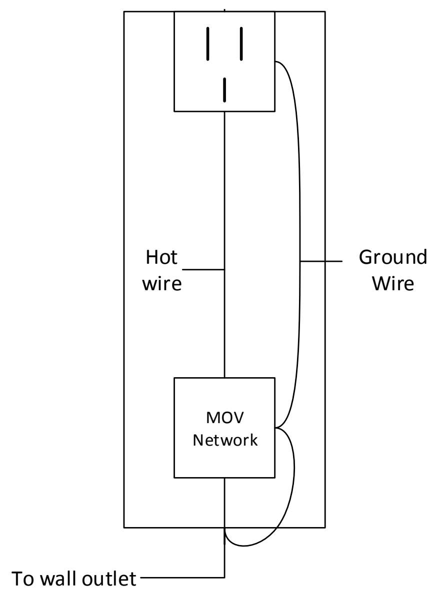


Figure 0-1.2-1: Surge Protection Basic

The TI Power strip offers protection using V430CH8T Varistors for the higher voltage sockets. The small elements are connected to each socket in the strip between the live wire and the ground. If power surges then the V430CH8T varistor will drain the excess power to ground protecting the devices. The voltage range of the V430CH8T is 14 to 275 volts AC according its datasheet. Damage will occur to the device for any voltages above 275 volts. This should be more than enough to offer protection in our circuit since voltage surges will most likely not exceed 275 volts. Figure 5.2-2 shows the circuit we will use for surge protection of our outlets. Each outlet including the power supply outlet will have the V4730CH8T varistors. Each outlet will have two of the components connected to the Live and Neutral part of the connector. Each varistor will also be grounded for the reasons given earlier.

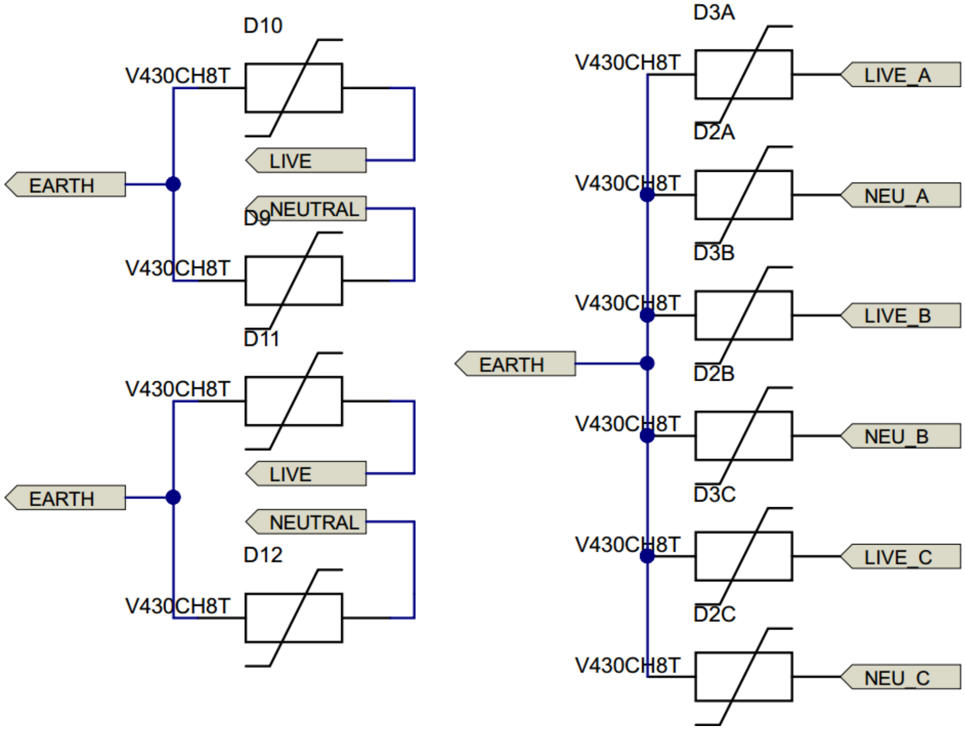


Figure 5.2-2: Outlet Surge Protection
Reprinted with permission from Texas Instruments

The design also uses Electro-Static Discharge (ESD) Transient Voltage Suppression diodes. These protective elements are used to offer protection for the general purpose I/O ports on the MCU. **Figure 5.2-3** shows how each ESD diode will be connected between the ground and any MCU input that will require a power to the MCU. These diodes will protect the MCU if any surges hit our VCC line. The diodes shown in the **Figure 5.2-3** have a max voltage of 15 V and max current of 6 A. Anything greater and the protection will activate and drain to ground. To keep costs down we may only use the protection for the main sockets. The protection elements on the MCU inputs will be used if we have the time and or money.

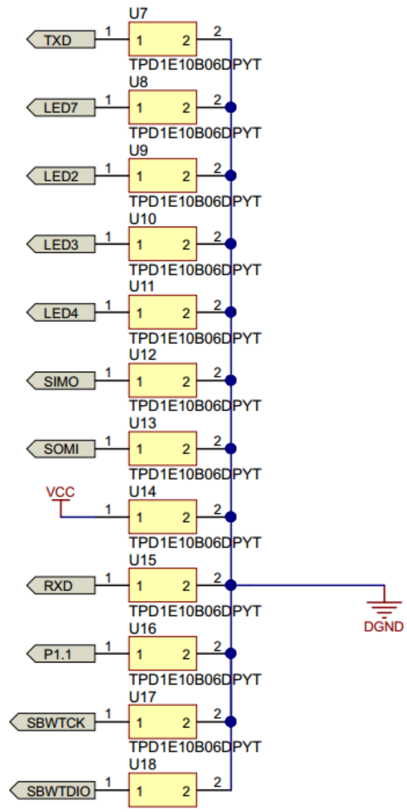


Figure 5.2-3: ESD protection.
Reprinted with permission from Texas Instruments

5.3 Power Monitoring

There are many ways to measure and record voltage, current, power, and energy readings, some more advanced and efficient than others. One of the more technologically unsophisticated and inaccurate ways to measure power consumption, is to read the power rating on the label of the electronic device. Almost every electronic device has the power consumption written on the label. Simply take this number (in Watts) divide it by 1000 and multiply it by the number of hours used to get the kilowatt hours (kWh) of the device. Most utility companies give prices in cost per kWh, so obtain what this cost is and multiply it by the amount of kWh used by the device and you'll get the price to run that device.

Another unsophisticated, but fairly more accurate and time consuming technique involves going to your electrical meter and counting the pulses on the meter. First, you would shut off all devices that consume large amounts of power and then go count the number of beats in a certain number of seconds. Take the beats multiply by 3600, divide by seconds, and multiply it by the kH reading on the electrical meter. This will give you the watts consumed when the device you're measuring is not plugged in. Then, you would plug in the device for which you want to obtain a power reading, and go out and count the number of beats in a certain number of seconds. Then repeat the process of taking the beats multiply by 3600, divide by seconds, and multiply it by the kH reading on the electrical meter. Then take your first reading and subtract it from your final reading and that is the watt usage of the device. Then, take this number divide by 1000 and multiply it by the number of hours used to get kWh.

Finally, there is the much more advanced and less time consuming method of actively measuring the power consumption of the device directly. There are many ways to do this. **Table 5.3-1** gives a few methods of measuring current and some of their capabilities.

Table 5.3-1: Common methods of measuring current

Type	**AC Compatible**	**DC Compatible**	**Intrusive**
Hall Effect Integrated Circuit Sensor	Yes	Yes	No
Current Clamp Meter	Yes	No	Yes
Resistive Sensor	Yes	Yes	Yes
Fiber Optic Current Sensor	No	Yes	Yes

The Hall Effect sensor measures the electromagnetic field around a cable and converts the signal to watts based on the strength of that electromagnetic field. The current clamp meter is used by clamping two jaws around two sides of an electrical conductor. This allows for the user to not have to disconnect the probe for insertion. The resistive sensor is achieved by measuring the power across a resistor; the resistor is of very small resistance, so that there is low power dissipation across it. The fiber optic current sensor only measure direct current and is done by wrapping a single-ended optical fiber around the current conductor and measures the interacting between light and the magnetic field around the conductor and converts to power. We will be using a resistive sensor for our power measurements, but more specifically we will be using current shunt monitors.

5.3.1 Current Monitor Selection

INA283 High-Accuracy, Bi-Directional Voltage Output Current Shunt Monitor

The INA283 is a voltage output current shunt monitor that senses the drop across a shunt in a voltage range of -14 V to +80 V. The supply voltage for this shunt monitor is anywhere from +2.7 V to +18 V. This component was designed specifically to measure voltages across current-sensing resistors, with very high accuracy. The INA283 has an effective bandwidth of 10 kHz. The Functional Block Diagram for the INA283 is shown in **Figure 5.3.1-1**.

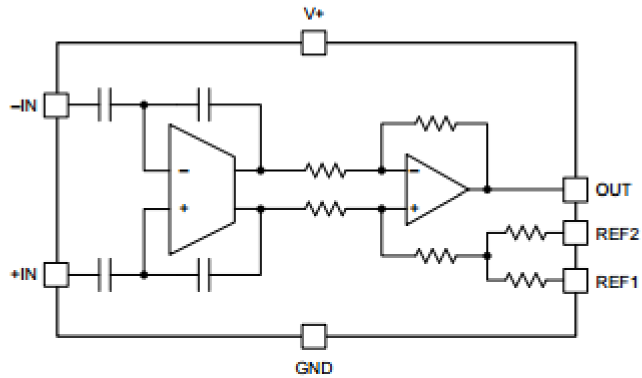


Figure 5.3.1-1: Functional Block Diagram for the INA283
Reprinted with permission from Texas Instruments

LMP8481 Precision 76-V Current Sense Amplifier

The LMP8481 is a high-side current sense amplifier that amplifies a small voltage differential across a shunt, called the input common-mode input range, in a voltage range of 4 V to 76 V. The LMP8481 is designed to be bidirectional. The supply voltage range of overlaps the input common-mode voltage range, so the LMP8481 can be powered by the input common-mode voltage range. This is nice, as it means there is no for an external voltage source. The LMP8481 has an effective bandwidth of 270 kHz. There is a very low-input offset voltage, so smaller sense resistors can be used, thus lowering the power dissipated across the resistor, without causing higher system error. The functional block diagram is shown in **Figure 5.3.1-2**.

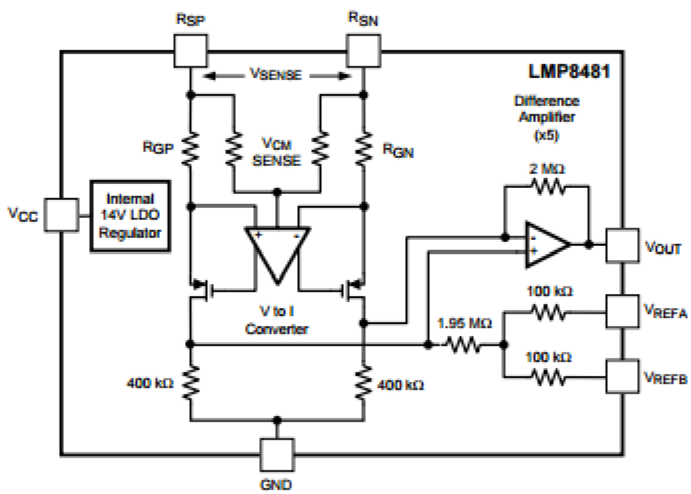


Figure 5.3.1-2: Functional Block Diagram for the LMP8481
Reprinted with permission from Texas Instruments

INA226 Bi-Directional Current and Power Monitor

The INA226 is a current shun and power monitor with a digital I^2C output. This is a nice feature because it eliminates the need for an analog-to-digital converter. Internal features, including programmable calibration value, conversion times, averaging, and an internal multiplier, allow for direct outputs of both current and power. The common-mode input voltage can vary from 0 V to 36 V. The supply voltage varies from 2.7 V to 5.5 V. The INA226 provides digital reading of current, voltage and power. The functional block diagram for INA226 is shown in **Figure 5.3.1-3**.

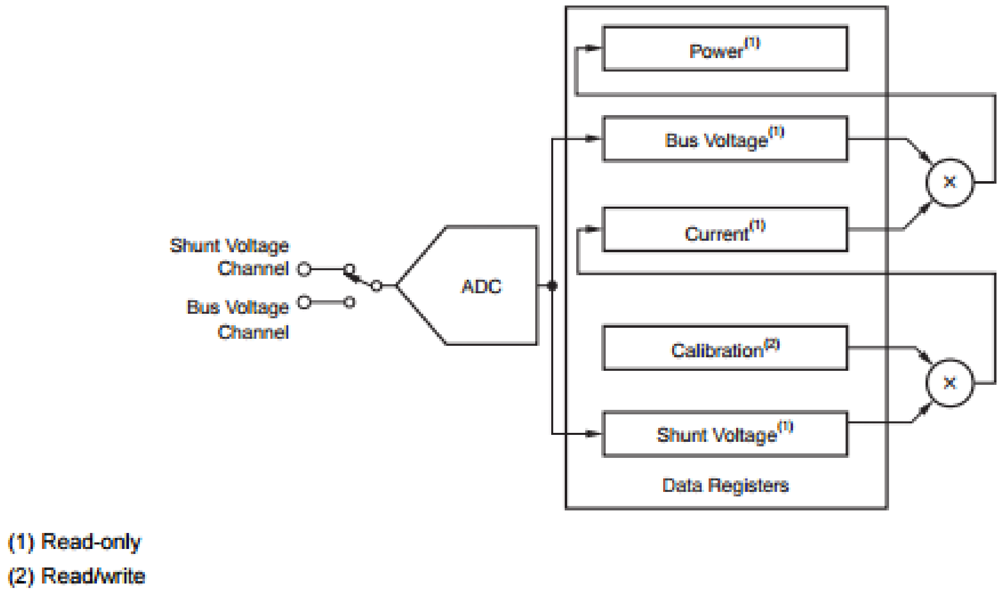


Figure 5.3.1-3: Functional Block Diagram for the INA226
Reprinted with permission from Texas Instruments

The INA226 Bi-Directional Current and Power Monitor was chosen for the current sensor that we would use. It was a hard decision, because the low common-mode input voltage is a bit of a concern, but it should work out fine. The fact that the chip reports current, voltage, and power as a digital output, was the deciding factor. We will be using three of these chips, one for each socket.

There are two measurements performed by the INA226 on the input power-supply bus. The load current develops a voltage that is sent through the shunt resister, creating what's known as a shunt voltage. This voltage is measured at the IN+ and IN- pins. The power supply can also be measured by connecting the voltage to the VBUS pin. This voltage is measured with respect to the ground, not the IN- pin. There is typically an external supply that is used to power the device. A full wave rectifier and resistive load will be used to step down the high input voltage before it goes into the device. The voltage measurements are converted to current, based on the Calibration Register value, and then power is calculated.

The INA226 operates in either continuous or triggered mode, and the mode has a direct effect on how the analog-to-digital converter works. The continuous operating mode is the normal operating mode. In the continuous mode, the shunt voltage reading and the supply voltage reading are continuously being read. The shunt voltage reading is then converted to a current value using the equation (1) below.

$$Current = \frac{ShuntVoltage \cdot CalibrationRegister}{2048} \quad (1)$$

These values for voltage and current are used to find the power, using equation (2) below.

$$Power = \frac{Current \cdot BusVoltage}{20{,}000} \quad (2)$$

Current is calculated based on what value is written into the Calibration Register. If no value is read from the Calibration Register, the current stored is zero. Power is then calculated based on the current and voltage. Thus, if there is no value in the Calibration Register, power is also stored as zero. An accumulator stores these values, and the chip continuously measures and calculates voltage, current, and power until the number of averages set in the Configuration Register (00h) is reached. For every sequence, the recorded values are added to the previously recorded values. After all averaging is done by the system, the final values for shunt voltage, supply voltage, current, and power are sent to the corresponding registers, where they can then be read. These values remain there until overwritten by the next completed series of results. The Conversion Register can also be set to where it only read shunt voltage or supply voltage, as to allow for more monitoring configurability to fit specific requirements. Conversion time is not affected by current and power calculations; they are done in the background.

The triggered mode on the other hand triggers a single-shot conversion. This allows for only one single set of voltage, current, and power measurements. The Configuration Register must be written to again, to trigger another single-shot conversion.

The device also has a power-down mode. This mode turns off all current into the device inputs, reducing the drainage of the supply voltage when the device is inactive. The registers can still be written to, even when the device is in power-down mode. This mode is active until the continuous or triggered mode settings are written to the Configuration Register.

The fact that INA226 does not directly measure current or power is an important part of the device. The device measures the differential voltage between the IN+ and IN- pin and the supply voltage applied to the VBUS pin. The Current Register (04h) and the value of the shunt resistor, to develop the differential voltage between the IN+ and IN- pin, are programmed, so that the device can report current and power ratings. The Power Register (03h) is set to be 25 times the Current_LSB, where Current_LSB is the programmed value for the least significant bit (LSB) for the Current Register (o4h). The Current_LSB and the shunt resistor value are used to calculate the Calibration Register value, which is then used to calculate current and power.

Equation (3) and (4) below are used to calculate the Calibration Register.

$$CalibrationRegister = \frac{0.00512}{Current\_LSB \cdot R_{SHUNT}} \quad (3)$$

Where:

- 0.00512 is an internal fixed value used to ensure scaling is maintained properly

$$Current\_LSB = \frac{Maximum\ Expected\ Current}{2^{15}} \quad (4)$$

5.4 Data (Diagrams Included)

This section will explain and describe the flow of the data in regards to the Power Strip sub system. There are two different scenarios in regards to the data flow with respect to the Power Strip. The first scenario will be the flow of the data to the Power Strip, and the second scenario will be the flow of the data from the Power Strip. The importance of noting the flow of the data is that the functionality of the Power Strip will be based on the data. Therefore, the data becomes a vital aspect of the functionality of the system as a whole. In the first scenario, the Power Strip is reading the power measurements, and the data is being displayed for the user on the web application, and it is being stored in the database for archiving reasons.

To explain the flow of the data beginning at the Power Strip, **Figure 5.4-1** begins at the power outlets that are built in the Power Strip. These are also referred to as the sockets on the Power Strip, and will be the means of connecting external devices to the Power Strip, and thus the system as a whole. Each of these sockets will be given an identification name or number by the user through the web application, in order to be able to differentiate between the sockets on the Power Strip. If nothing is plugged in to the socket, the user can then delete the socket from the web application. The electricity will flow from the outlet on the wall to each of the outlets on the Power Strip. This is not yet seen as data in the system, but will be converted to values that calculations can be performed on once the flow progresses to the Power Reading Hardware stage.

After the electricity flows to the outlets on the Power Strip, the Power Reading Hardware will convert the flow of electricity into measured values that can be used to perform calculations on how much power the user uses in a specific socket, or in the Power Strip as a whole. The Power Reading Hardware will keep track of the power usage, and the objective becomes displaying this data to the user, and storing the data in the database to be displayed for the user when necessary. This data will then be read into the microcontroller that controls the Power Strip. Once the data is in the microcontroller, the data will be transferred over to the web server over Wi-Fi. The server will then react to the data received from the microcontroller by storing the data in the corresponding data in the database. After this data is stored in the database, it will then be pulled from the database to be displayed in the web application. This shows how the web server will serve as the central hub of the system because the data in relation to the Power Strip will have to go through the web server and into the database, and then back to the web server and to the web application. This is opposed to sending the data directly from the microcontroller to the web application. This method will prevent data redundancy, and will provide an efficient method to transfer data.

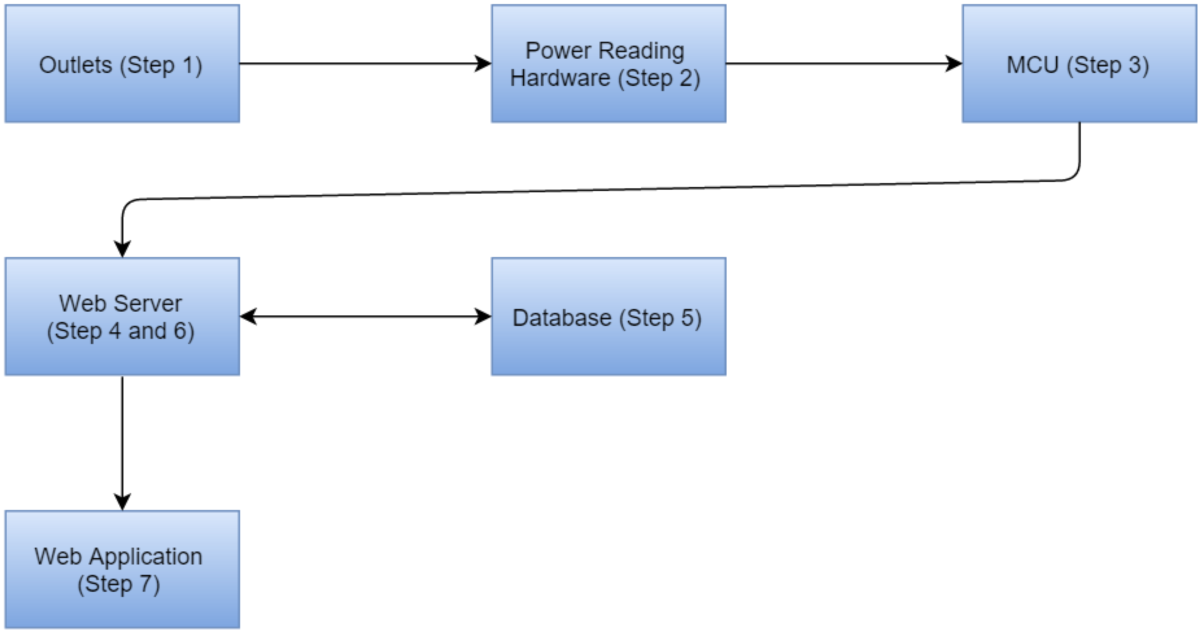


Figure 5.4-1: Data Flow Starting at Power Strip

Similar to the first scenario, in which the data flow starts at and exits the Power Strip, the second scenario will have the data flow end at and enter the Power Strip. In this second scenario, the data will began at the database because any data that impacts the functionality of the Power Strip will be based on data that has already been persisted in the database. As shown in **Figure 5.4-2**, the data flow will begin in the database and will be pulled to the server based on the request sent to the server. This request is fired when the user changes settings in the web application, and submits the settings. The settings are then saved to the database, and retrieved again from the database. Once the web server retrieves the data from the database, the data will then be sent to the microcontroller on the Power Strip. After the microcontroller receives the data from the web server, it will perform actions and change the functionality based on the data received from the database. This may include shutting off power in specified sockets on the Power Strip. Therefore, the flow will continue to the outlets, skipping the Power Reading Hardware that was a part of the flow to describe the first scenario.

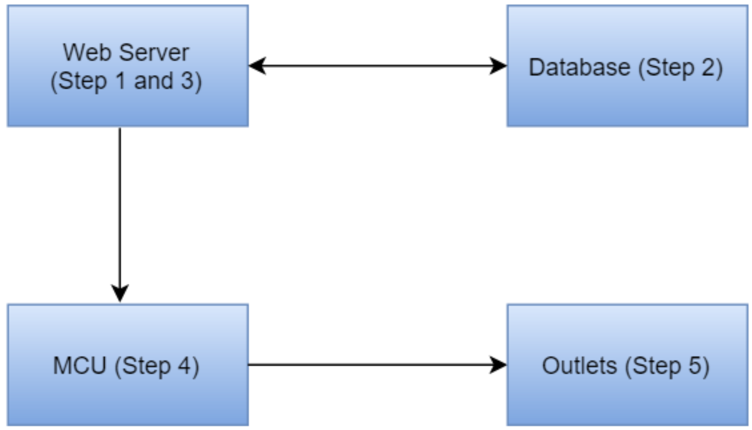


Figure 5.4-2: Data Flow Ending at Power Strip

5.5 Microcontrollers

When choosing a microcontroller for our design, we had to come to a firm conclusion about our general requirements, after reviewing our system, we decided that our choice needed to be energy efficient, to be able to support at least four peripherals, and to be able to connect to Wi-Fi. A final decision was also made to choose one microcontroller to be present in all devices to simplify design considering that we would only have approximately four months of implementation, and having to use different microcontrollers with different architectures could make out design overly complicated given that the project, with an already tight deadline, with such diversity in hardware would make it difficult to accomplish the task at hand.

The two options which we have chosen to analyze is TI's MSP432, a general purpose microcontroller with large amounts of memory, many project-relevant libraries, high input flexibility, and a robust design, and the CC3200, a first of its kind, internet on a chip microcontroller, with limited memory space and a more focused design, being able to supply its own access point or latch on to others available to the environment with built-in Wi-Fi drivers and hardware. If these requirements can be met at a low cost with peripherals then that would also be acceptable. Cost, not performance, was the main factor in coming up with our decision.

5.5.1 MSP 432

The Texas Instrument MSP432 is the newer more energy efficient model of its predecessor, the MSP430, which all teammates had had experience working with in a course on embedded systems. It is a widely used model designed to be robust in its design in order to be able to solve a wide variety of problems. With a large number of options available to it in terms of both functionality and peripheral tools, its cost and power usage is very low, with TI bragging that it can run for an entire year on the same amount of power supplied by two AA batteries.

A major focus of this microcontroller's design is the ability to accept many varying peripherals, and to ensure that the correct information is being transferred between peripherals and the microcontroller, TI has implemented a very low leakage input/output bus design. The MCU's low power usage and the familiarity of teammates with its design makes it a very tempting choice but with the high cost of peripherals, further in depth considerations must be made considering our limited budget. The general features of the microcontroller design and hardware design include:

- Ultra Low leakage I/Os
- A Floating-Point Unit
- Independent Flash Banks
- Runs in active mode at 95uA/MHz
- Real Time Power Measurement and Debugging
- 256-bit AES Encryption
- 256 KB of Flash
- 64 KB of RAM
- 32-bit ARM Cortex M-4F
- 48 MHz CPU

5.5.1.1 Tools

As well as the hardware, the software tools available to developers or provided with the design can be just as, if not more, important in increasing a device's efficiency. Developed by professionals with years of experience refining some of the drivers, API, and IDEs listed below, software provided by the manufacturer is more beneficial to purchasing developers, because of the finesse and expertise already applied to ensuring a well optimized product.

On the MSP432 there are many drivers and libraries available to the use of developers to assist with wireless inter-device communication, efficient usage of memory busses, and optimized peripheral-microcontroller operations.

5.5.1.1.1 MQTT Library

The MQTT Library provides the developer with streamlined, built-in C/C++ functions which assist in the maintaining, initiating, and upholding of the overall MQTT communications protocol. Short for the message queuing telemetry transport protocol, MQTT is a data light messaging protocol to manage TCP/IP addressing protocol. Based on a publisher to subscriber model with one central hub, called the broker, the broker is needed to disperse information to many devices when changes occur to information or in our case, table which the developer is subscribed to. The benefit of MQTT protocol is it is a light weight messaging model which was designed for embedded systems and internet of things devices, which is exactly what our project is classified as. This light weight protocol is perfect for systems with low amounts of memory which is limited in the amount it can allocate towards tasks such as digital communication between low memory embedded systems and internet of things devices.

The responsibility of developers with this library is to utilize to set up subscriptions between the data supplied by the broker, and the relevant devices, to have the broker actively alert the necessary subscriber devices whenever a change in the device occurs, and for the broker to maintain and process the network and network requests. Many optimized functions have been provided by this memory, created by experienced and professional developers to assist in a wide variety of MQTT communication scenarios for developers using Texas Instruments' MSP432 coupled with a peripheral to ace the internet.

5.5.1.1.2 SimpleLink Library

Providing a simple set of APIs to fit the needs of most applications on the MSP432, the SimpleLink Component Library, displayed in **Table 5.5.1.1.2-1** assists in managing devices, wireless network configuration and setup, while handling Berkley sockets.

Table 5.5.1.1.2-1: The responsibilities covered by each section of the components of the SimpleLink library Reprinted with the Permission of Texas Instruments

Components	Functionality
device	• Initializes the host • Controls the communication with the Network Processor
wlan	• Connection to the access point • Scan access points • Add/Remove access point profiles • WLAN Security
socket	• UDP/TCP Client Socket • UDP/TCP Server Socket • UDP/TCP Rx/Tx
netapp	• DNS Resolution • Ping remote device • Address Resolution Protocol
netcfg	• IP/MAC address configuration
fs	• File system Read/Write

The SimpleLink's socket API is the most commonly set to assist in managing user applications by managing UDP/TCP receiving and transmitting protocol, UDP/TCP client sockets, and UDP/TCP server sockets while adhering to the Berkley standard device socket API. The FS API provides access to the serial flash memory components of the device MCU. SimpleLink's APIs are required to uphold the wireless system's maintenance and operation to uphold internet connectivity. It Manages start, get, stop, and set device configurations, regulating inter-device communication by establishing host initialization and communication with the device's built-in network processor.

While maintaining wireless local area network and general connectivity related protocol such as 802.11 and available access point connections, the WLAN API also sets method provisions, managing device connection modes, including station, access point, and peer-to-peer Socket APIs. The SimpleLink library components also support dynamic and static memory models, and sets connection policies, while adding and removing network and access point profiles, and constantly scans the environment to find more access points to connect to while idling.

The network protocols managed by the SimpleLink library ensure that continued communications can be performed between devices. It enables different networking services, upholding various connectivity-based protocols, MDNS client/server services, DHCP server services, and HTTP server services, the NetApp API also stores address resolution protocols important for continued connection between devices, DNS resolution, and the pinging of other remote devices.

The library's network-oriented APIs are necessary for upholding network functionality by managing the device networks resolution and configuration. And the network configuration component of the library configures network parameters by acquiring and setting static IP addresses through DHCP to MAC addresses, ensuring that communications to the proper devices are upheld, while offering read write operations for these functions to ensure that IP protocol is maintained. The SimpleLink library components also support dynamic and static memory models.

5.5.1.1.3 DriverLib Library

The architecture of the hardware was designed to closely incorporate Texas Instruments' DriverLib Library with the central processing unit uses a long list of APIs to reduce the MSP432's flash footprint, giving flash memory more space to devote to other tasks, adding pathways to implementing both initialization and control functions of on-chip peripherals. The driver library was also designed to optimize memory usage, improve solution robustness, and reduce development time by extensively using the available APIs offered by the library. With a long list of drivers provided for the MSP432, the DriverLib library's role to optimize bus utilization, reduce data traffic, and bring further efficiency to microcontroller and peripheral interactions can be more effectively accomplished.

5.5.1.2 Memory

The MSP432 supports a large amount of memory for an MCU, four gigabytes of memory, divided into six sections, a code zone, SRAM zone, peripherals zone, debug and trace peripheral zone, and a large unused zone. All memory is addressed between 0x0000 0000 and 0xFFFF FFFF. Many of the zones mentioned above are also embedded in other zones, keep in mind if a zone of the same name is mentioned twice but has a different prefix number it may also have a memory enclave in other zones. Each zone's functionality and organization, featured in **Figure 5.5.1.2-1**, will be covered in detail below, and addresses which are not mentioned below are reserved for other functions**.**

Figure 5.5.1.2-1: A general overview of the MSP432 memory layout
Reprinted with the Permission of Texas Instruments

5.5.1.2.1 Code Zone

Accessed through the ICODE and DCODE memory busses of the Cortex M-4 processor and DMA, the addresses between 0x0000 0000 and 0x1FFF FFFF is referred to as the code zone. Contained within this address band is flash memory, ROM and internal SRAM, which is where a large majority of data used for regular operation, data storage, and quickly accessible application data is stored.

5.5.1.2.1.1 Flash Memory

Flash memory is used for external storage to hold data that is needed but necessary in use commonly during the operation of the MCU, such as unused application code which is initiated through exception and interrupt calls. Addressed from 0x0000 0000 to 0x003F FFFF, a four megabyte region, the MSP432's low power, high endurance flash memory is one-hundred twenty-eight bits wide, and is able to support up to twenty-thousand read and erase operations. The bit width is highly beneficial for storage because it is able to hold up to four thirty-two bit instructions per segment, lowering the number of instruction fetches necessary from the flash when utilizing power efficient the burst write operations. The subsections of information memory and main memory is stored in flash, and are divided so that an application can read and write into one section while the other section is executing erase or program operations simultaneously.

Consisting of sixty-four four kilobyte sections, the two-hundred fifty-six kilobyte section is also divided in half like the entirety of the flash for the same reason flash memory was divided, to be able to alter memory from one section while another section is executing a program or erase operation. The second major section of the flash, information memory, is sixteen kilobytes made up of four four-kilobyte segments, which is also divided in half to take advantage of the same functionality of flash and main memory. Both the information memory and main memory subsections have the ability to perform mass erase and mass write protection ensuring that important data is safeguarded from such functions. When experiencing brown outs or other low power occurrences flash memory, to prevent leakage is set to a power down state.

5.5.1.2.1.2 SRAM Memory

The SRAM memory region is present in the code zone as well as having its own SRAM zone to allow for time efficient access to SRAM, deterring the device from making large jumps between memory addresses for both data reading and instruction fetches, but instead utilizing single cycle memory fetches using the pipeline protocol. The task of SRAM is to store memory which is accessed for read and write operations very often by the MCU, and is usually associated with a currently running application. This specific section of SRAM functions in the same way as a memory buffer does, located in a different section of the architecture to facilitate quick access to heavily used information instead of having to make multiple time consuming jumps to the more distant SRAM section of memory. This section of SRAM is present to optimize runtime of various applications on the microcontroller. More information about SRAM’s organization and functionality will be covered in the SRAM zone section.

5.5.1.2.1.3 ROM Memory

From 0x0200 0000 to 0x020F FFFF, also known as the ROM region, is only thirty-two kilobytes of utilized memory with the remaining one megabyte reserved for further TI upgrades and patches. The Rom will hold the boot loader and various drivers assisting in the handling of processor peripherals. It Stores data important to the function of the system such as the operating system which is integral to start-up, interrupt and exception handling, and the very basic function of the device. This data cannot be written over without causing catastrophic damage to the entire system, with its effects potentially bringing the entire operation of the MCU to a halt. Other essential programs usually stored in ROM also includes driver software to handle low level hardware operations, ensuring that the entire MCU runs smoothly and while efficiently utilizing all of the hardware’s resources.

5.5.1.2.2 SRAM Zone

Static random access memory is able to store and remove data that is used for an application for a short period of time. Sorting data which is quickly read from and written into, this form of memory fetches instructions for the application until it terminates or an application of higher priority begins, stopping the lower priority code through an interrupt or some similar process. Random access memory can be altered or rewritten, is used in storing data from basic calculations and is extremely important for all aspects of computing.

The general layout of the memory in the SRAM zone is featured in **Figure 5.5.1.2.2-1**.Addressed from the range of 0x2000 0000 to 0x3FFF FFFF is divided into two sections, the SRAM memory region, addressed from 0x2000 0000 to 0x200F FFFF, and the SRAM bit-band alias memory region, addressed from 0x2200 0000 to 0x23FF FFFF will be covered below. The MSP432 supports up to sixty-four kilobytes of SRAM. The SRAM, like the flash memory can also be put into a low leakage low power state, ensuring that the SRAM can be relied on regardless of power states.

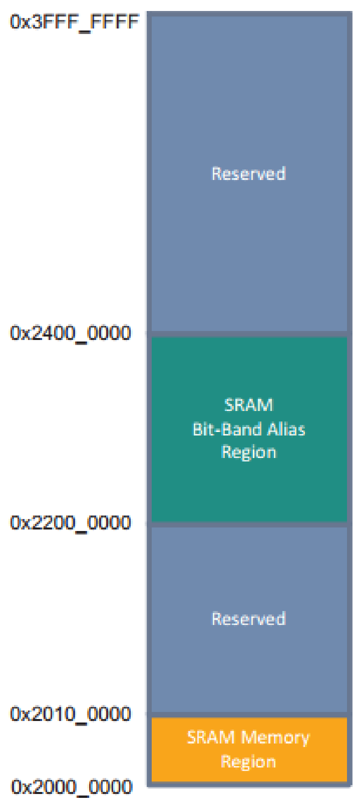


Figure 5.5.1.2.2-1: The memory layout of the SRAM Zone
Reprinted with the Permission of Texas Instruments

SRAM power can be optimized by applications which are given the choice to power down individual eight kilobyte blocks of SRAM that will remain powered down when entering and reemerging both to and from active and low power modes. This ability limits any unnecessary currents when the MCU switches states between power modes. This feature offers a wide array of flexibility for the developer, letting them have a block powered up in separate stages of processing while leaving it powered down for others according to the developer's discretion, and knowledge of the amount of memory space needed at each stage.

Disabled blocks of memory cannot be written into and will return a value of ‘0h’ when read from. To prevent these holes from causing issues during a program’s execution, protective measures were taken by the designers of the MCU, all powered down lower addressed blocks are powered up when a higher address block is powered up. This procedure is very important to note, and should influence the developer’s choice to power down blocks in the order of lowest memory address to highest memory address, similar to a stack, for least likely to be powered up to most likely to be powered up memory segments, to best utilize the power saving properties of this tool.

5.5.1.2.3 Peripheral Zone

From range 0x4000 0000 to 0x5FFF FFFF it is divided between the peripheral memory region and the bit-banded alias peripheral memory region. The one megabyte peripheral memory region is dedicated to the device’s system and application control peripherals which are mainly drivers devoted to the operation of additional features present in the MSP432. The bit banding alias peripheral region is reserved for bit-banding, a feature provided by the Cortex M-4 processor which allows for atomic bit alterations of memory addresses while also avoiding the need to utilize the pipeline bandwidth. Bit-banding increases the efficiency of the MCU, allowing other processes to take advantage of the benefits of the pipeline protocol by surpassing the pipeline process all together.

5.5.1.2.4 Debug and Trace Peripheral Zone

Spanning 0xE000 0000 to 0xFFFF FFFF of the memory, the main responsibility of the debug and trace peripheral zone is to map the internal and external private peripheral bus regions of the Cortex M-4 which are comprised of the core and system debug control registers, NVIC registers, other general system control registers, the system controller, reset controller, and the debug ROM table.

5.5.1.3 Direct Memory Access

The eight channel ARM DMA, with its abstract architecture available in **Figure 5.5.1.3-1,** allows for eight simultaneous data transfers to occur between memory and peripherals without relying on the bandwidth of the CPU, this saves power by allowing the CPU to revert to an idle state while eight or less memory to peripheral data transfers are initiated. The DMA can also remain active in multiple low power modes, further idling the CPU, lowering the number of wake-up interrupts called for the CPU, and the DMA can generate four of its own interrupts without the assistance of other drivers or peripherals, allowing for more hardware components to remain inactive while in a low power state.

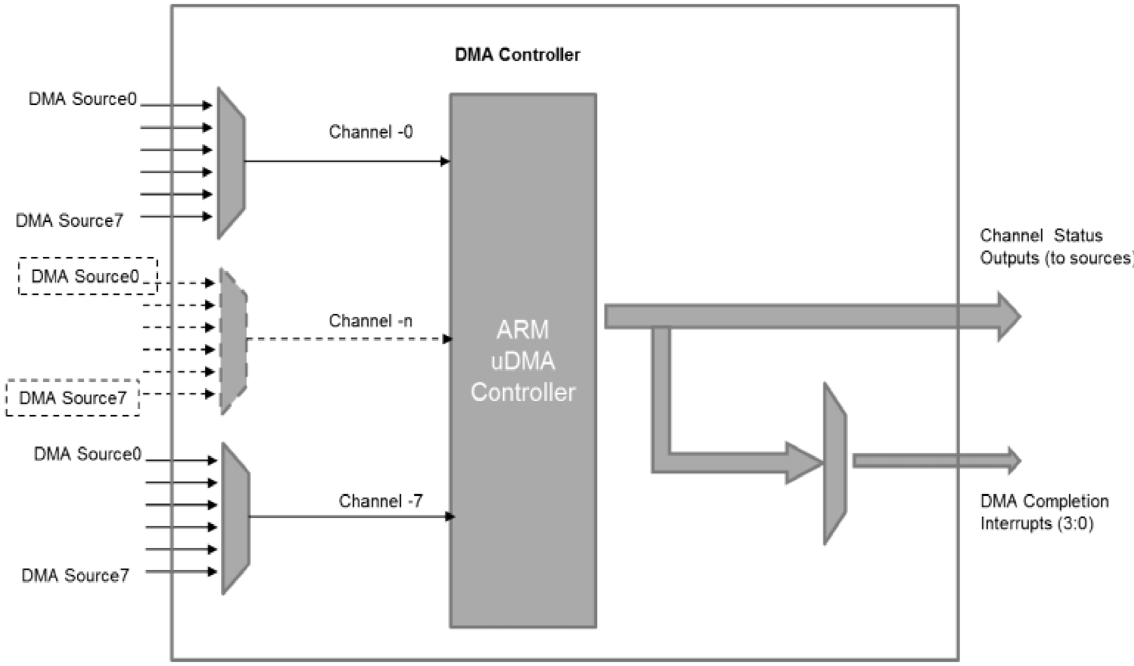


Figure 5.5.1.3-1: An abstract diagram displaying the physical components of the MSP432's direct memory access hardware design.
Reprinted with the Permission of Texas Instruments

The DMA has access to all memories and peripheral configuration interfaces with the exception, for security purposes, of the upper bank of the flash's main and information memories in the event that the DMA bus could be used as an unauthorized access source to configured IP addresses, and possible store or alter these addresses to either learn where the device's communications are being sent or to send the device's communication to an unauthorized device. The direct memory access controller is a complex piece of hardware which utilizes the architecture of the MCU to most efficiently use memory space to store and retain data during times of low power, minimize processor usage, and fully utilize bus bandwidth by transferring data from the processor, and performing transfers of data between memory and the peripherals of the MCU.

5.5.1.4 Processor Details

The Arm Cortex-M4 is central to the operation of the entire microcontroller, as a high performance, low cost processor, with a small pin count, this low power, and low memory model to runs efficiently when applied to the functionalities of the MSP 432's architecture. The wake up interrupt controller makes it possible for the microcontroller unit to enter low power sleep states quickly, and awake from them, which is especially useful for timely, efficient processing, when devices experience long periods set in an idled states.

Closely integrated with the core processor the low latency interrupt software, the M-4 Cortex is designed to optimally perform within embedded systems and internet of things devices, while utilizing a nested vectored interrupt controller by fetching the necessary interrupt, saving the processor state in a parallel, tail chained interrupt, prevents unneeded save states from occurring in instances when multiple interrupt calls have been made, allowing for efficient interrupt processing for late arrivals. Three advanced high performance bus interfaces, the ICode, DCode, and system bus interfaces, allow for faster, efficient processing of different data types, and also features atomic read and write functions.

5.5.2 CC3200

When choosing a microcontroller for our design, a few factors had to be examined, cost, performance, and usability. The Texas Instruments' CC3200 is a standalone microcontroller with Internet-on-a-chip capabilities. The first of its kind, it is able to connect and communicate with servers to manipulate databases through an external access point or, instead, generating its own. With built-in Wi-Fi connectivity, security features, and helpful APIs, past experience is not required to ensure quick, relatively hassle-free development. Internet-on-a-chip devices are especially practical for customers on a budget, allowing them to purchase a single device which provides the full range of connectivity options available without having to purchase or rely on the full functionality of home wireless networks. Having the ability for MCUs to communicate without being dependent on an access point also provides enhanced stability to the system. If a wireless network unexpectedly becomes inoperable with this microcontroller only a power source is needed to ensure full functionality. With the extensive amount TI example code provided through their CC3200 SDK, our familiarity with coding other CC3200 microcontrollers, online example videos, and their documentation outlining the requirements for starting up and making the microcontroller fully functional, we decided a TI microcontroller was the best choice for our design. Eventually we decided on the CC3200 MOD shown in figure 8. We decided on this package for its ease of Wi-Fi hardware and easy to read reference designs for PCB implementation. The MOD is well documented so it was easy to research and implement.

The Texas Instruments' CC3200 is a standalone microcontroller with Internet-on-a-chip capabilities, the first of its kind on the market, allowing it to connect and communicate with databases and other microcontrollers without the necessity of an external access point, supplying its own instead. With built-in Wi-Fi, Internet, encryption and security protocols, past experience is not required to ensure fast development and functionality. An internet-on-a-chip device is especially practical for customers on a budget, allowing them to purchase a single device while still providing the full range of abilities available on each individual device without having to purchase the entire set or rely on other, sometimes unreliable, wireless networks. Having the ability for MCUs to communicate without being dependent on an access point also provides stability for the system if a LAN router or another wireless network device unexpectedly becomes inoperable. With the microcontroller only a power source is needed to ensure our design is fully functional. The general featured by this microcontroller include:

- 802.11 b/g/n station and access point, fully integrated radio, baseband, and MAC

- Single-chip Wi-Fi MCU: Wi-Fi Network Processor and ARM Cortex-M4 MCU on one chip with RF design
- TCP/IP stack and Wi-Fi Driver transparent to applications
- Embedded Crypto Engine for security
- Low Power and Advanced low power modes
- Embedded hardware crypto engine for TLS/SSL internet security
- 250 kB of RAM
- ARM Cortex-M4 Core
- 80 mHz CPU
- Wifi and internet protocols in ROM
- Micro Direct Memory Access

5.5.2.1 Tools

The tools offered by a microcontroller's libraries drivers, etc. can impact the entire project's design immensely. If libraries are present focusing on optimizing the tasks performed by a given product, a team of developers can guarantee a minimum standard of quality similar to that of the professionals who designed many of these built-in functions, knowing that a sizable portion of their program utilizes these functions. Libraries can vastly improve the time needed to implement a project, simply due to the fact that many functions integral to the operation of the system are already preprogrammed by experienced programmers working for the microcontroller's manufacturer. Drivers offered by the design further increase the microcontroller's efficiency, ensuring that the optimal use of buses is achieved between many complex architectures, even down to the lowest level of design.

5.5.2.1.1 SimpleLink Component Library

The SimpleLink Component Library assists in streamlining basic device management, wireless network configuration and setup, and BSD socket handling while providing a simple set of APIs to fit the needs of most applications on the CC3200.

The various APIs have different responsibilities and functionalities important to maintain the operation of device interconnectivity. The Device API manages start, get, stop, and set device configurations, mostly concerned inter device communication by establishing host initialization and controlling communication with the network processor. While maintaining WLAN related protocol such as 802.11 related-functionality and access point connections, the WLAN API also sets method provisions, manages device connection modes which include station, access point, and peer-to-peer Socket APIs, sets connection policy, adds and removes network and access point profiles, and constantly scans the environment to find more access points to connect to in its environment. The Socket API is the most common API set for user applications, managing UDP/TCP receiving and transmitting, UDP/TCP server socket, and UDP/TCP client socket while adhering to BSD socket API. The FS API provides access to the serial flash memory components of the device MCU.

The network-oriented APIs are particularly important for maintaining network stability and functionality, managing the resolution and configuration of the device's network. By enabling different networking services such as HTTP server services, DHCP server services, MDNS client/server services, and upholding various connectivity-based protocols, the NetApp API also regulates address resolution protocols, DNS resolution, and the pinging of other remote devices. And the NetCfg configures network parameters by setting static IP addresses, acquiring IP addresses through DHCP, and setting MAC addresses, ensuring that communications to the proper devices are upheld. And by offering the read write operations of user proprietary and networking data.

In our design, we specifically used APIs present in the HTTP_Client libraries, utilizing the GET and POST functions provided to enable server client communication with the website. Within these functions a handshake is established, and received messages are parsed through the JSON library, responsible for parsing JSON responses. To establish connection through a local access point, we used the simplelink library's API which acquires an IP address in order to properly communicate with the server by using the aforementioned APIs.

Also supported are static and dynamic memory models, SPI communication channel access functions, and non-OS/single threaded and multi-threaded OS environments. But special considerations, such as access protection and synchronization, must be accounted for and implemented by the developers to ensure the reliability of a multi-threaded OS on such simple hardware. Switching between OS modes is able to be achieved simply by utilizing the SimpleLink IDE workspace.

5.5.2.1.2 Peripheral Driver Library

Integrated as part of the MCU's ROM, TI's DriverLib uses a long list of APIs to reduce the CC3200's flash footprint, allowing flash more space to devote to other tasks, providing pathways for implementing both initialization and control functions of on-chip peripherals. The driver library was also designed to optimize memory usage, improve solution robustness, and reduce development time by extensively using the available APIs offered by the library. The built-in APIs supported by the driver library include assistance with:

- Analog and Digital Conversion
- Advanced Encryption
- Camera Handling
- Redundancy Checks
- Flash Memory Handling
- General Purpose I/O Pin Handling
- Inter-Integrated Circuit Handling
- Inter-IC Sound Protocol Handling
- Interrupt Handling
- Pin Handling
- Power Reset Clock

- Securing Digital Host
- Serial Peripheral Interface
- General Purpose Timer
- Universal Asynchronous Receiver/Transmitter
- Direct Memory Access
- Watchdog Timer Handling

5.5.2.2 Pin Layout

The CC3200 has a 64-pin QFN system which allows for extensive pin multiplexing as well as Texas Instruments' recommended pin layouts for specified functions. A diagram of all the pins available in the CC3200 are displayed in **Figure 5.5.2.2-1**. The pins will be covered in more detail in terms of use and function in the following sections.

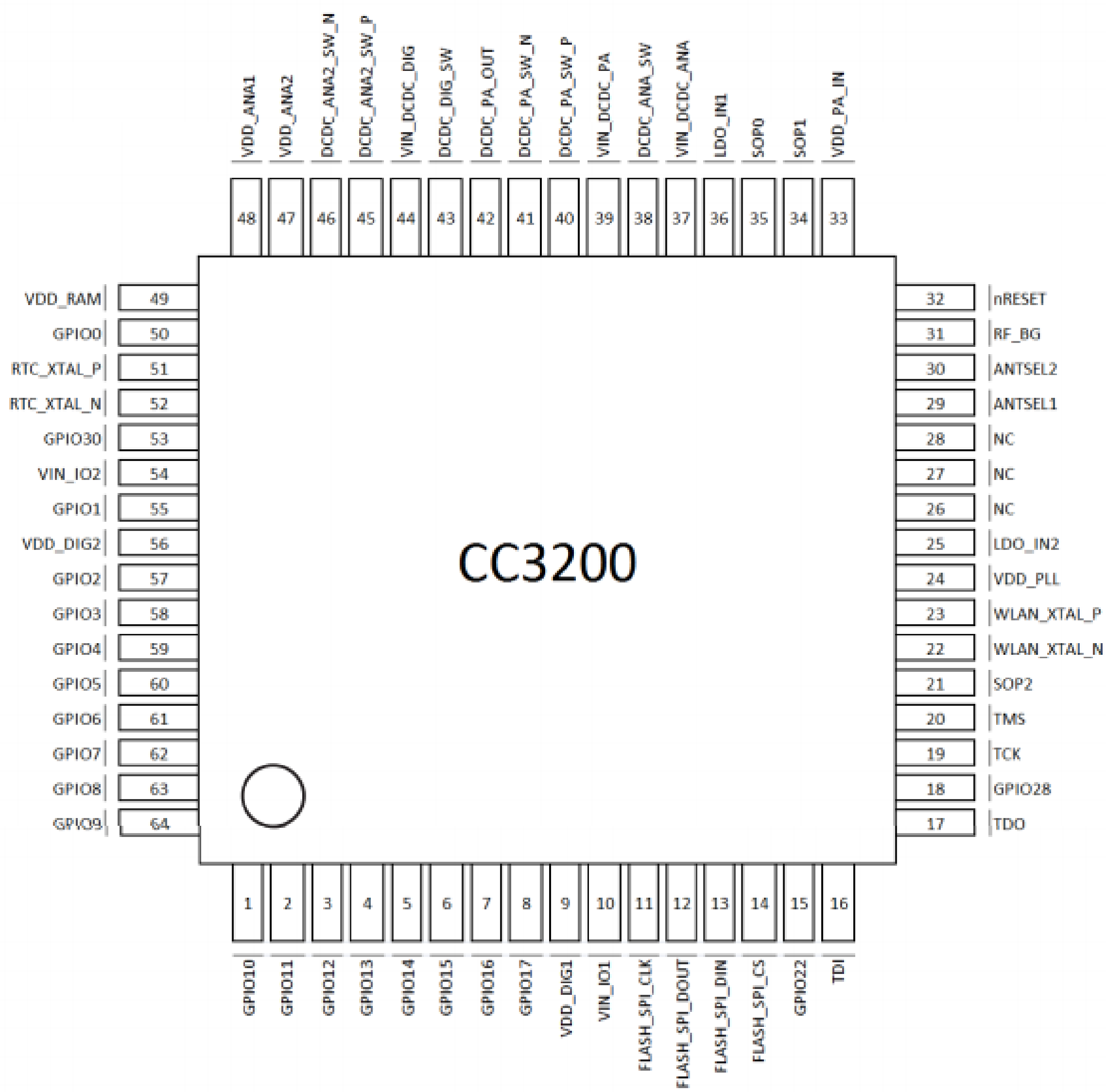


Figure 5.5.2.2-1: The visual diagram of the CC3200's physical pin placement
Reprinted with the Permission of Texas Instruments

5.5.2.2.1 General Purpose Input/Output Pins

Up to 24 pins can be used as general purpose input/output pins, and each pin of this type is grouped in four eight-bit modules. All pins are interrupt capable, can be used to trigger DMA operations, and are all general input output pin register readable through high-speed internal bus operations. This pin type is designed to accommodate communications and data transfers from the microcontroller's main memory to its various peripherals and peripheral memories. We used the GPIO pins assigned to the LEDs to regulate relay control in our design.

5.5.2.2.2 Analog to Digital Converter Pins

Some pins are designated to handle both digital and analog inputs, having the ability to convert analogous voltages into discrete digital numbers, designed as twelve-bit, four analog input channels, using automatic round robin sampling it has access to dedicated direct memory access channels to transfer their data directly to the application RAM with an effective nominal accuracy of ten-bits. This is especially useful when using peripherals which only transmit analogous data directly to the microcontroller. In this case, the microcontroller must connect the peripheral output to these channels to be properly processed.

The analog to digital converter pins can measure up to 1.8V each, but to prevent damaging the microcontroller it is recommended that the values do not exceed 1.466V. Having a 12-bit resolution, each pin can accurately measure voltages in increments of .00035V which allows our design to accurately measure .04 W for each appliance. There are four analog to digital converter pins in our design. With one pin measuring voltage and three measuring current, this allows for three sockets to be independently measured in our design.

5.5.2.3 Memory

Necessary for the entire functionality of the design, memory space, memory availability, memory organization, and efficient memory utilization play heavy roles in determining whether or not the CC3200 is a satisfactory choice for our design. An in depth analysis of these features is necessary, as well as analysis of the many structures working alongside memory dealing with data transfer. **Figure 5.5.2.3-1** gives a visual representation of the CC3200’s physical memory layout which will be covered in detail below.

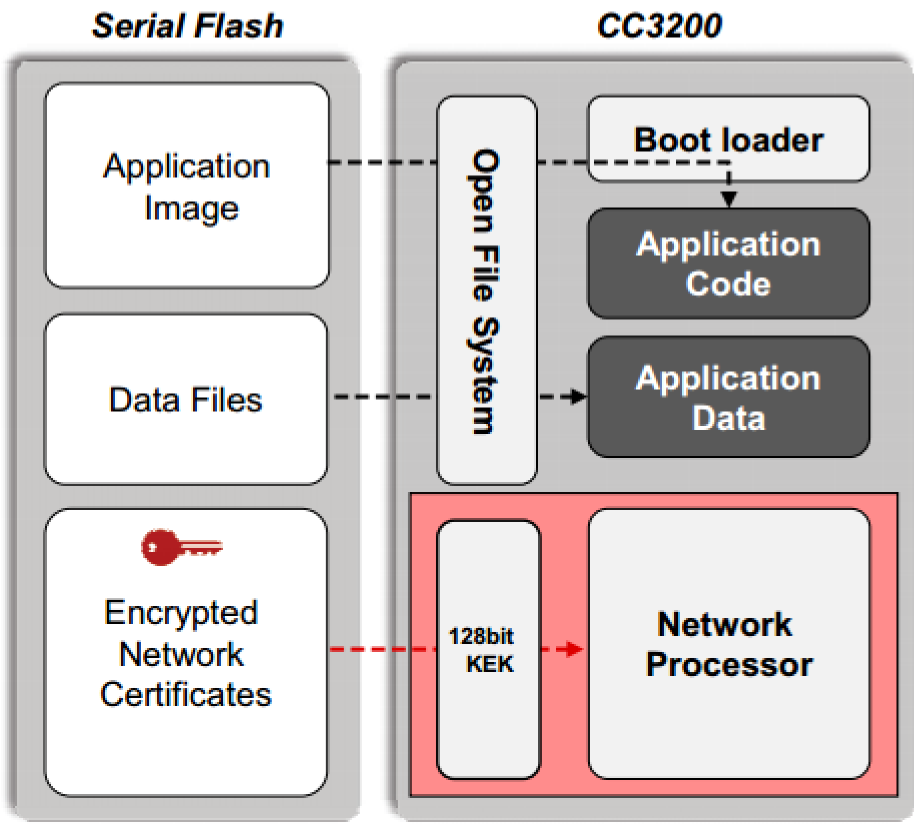


Figure 5.5.2.3-1: A diagram of the CC3200's memory and memory usage, displaying inter-memory connections and functions.
Reprinted with the Permission of Texas Instruments

5.5.2.3.1 Flash Memory

The external memory is memory which is included with the CC3200, but is not part of the actual microcontroller, it contains only flash memory which contains data that does not need to written or read from often when performing tasks required for the microcontroller's operation, and stores data related to CC3200 driver and built-in software patches and updates from Texas Instruments. Also, the external flash memory acts as off-MCU ROM, with greater space, 512KB, it stores all service pack files, certificate files, configuration files, system files, user files, and web page files. The space allocated for flash memory can be adjusted using an API format command, with recommended sizes of each section to ensure fail-safe and non-fail-safe storage in **Table 5.5.2.3.1-1** below.

Table 5.5.2.3.1-1: The recommended space used in flash memory to accommodate specific functions, divided in to fail-safe and non-fail-safe storage options.
Reprinted with the permission of Texas Instruments

Item	Typical Fail-Safe	Typical NonFail-Safe
File system	20KB	20KB
Service pack	224KB	112KB
System and configuration files	216KB	108KB
MCU code	512KB	256KB
Total	8Mb	4Mb
Recommended	16Mb	8Mb

Files are stored in the order that they were downloaded, and are therefore never stored in a fixed location. So, files cannot be called directly from a direct location in flash memory, and can only be accessed through their human readable file name.

All flash blocks are 4KB, and therefore all files take up a multiple of 4KBs of space. With 512KB total, a maximum of 128 files can be supported, and encrypted files with fail-safe support are double their original size block size, halving the number of supported files to 64.

5.5.2.3.2 RAM

The benefit of random access memory is its ability to quickly store and remove data which is only needed for a temporary period of time. This data is disposable and is used primarily to store and write data, and fetch instructions for various applications until either the application terminates or a higher priority application initiates through an interrupt or some similar process. Random access memory can be altered or rewritten, is used in storing data from basic calculations and is extremely important for all aspects of computing.

The CC3200 holds 256KB of zero-wait-state, on-chip SRAM, located at memory locations 0x2000 0000, which is shared between downloaded code and data. With the thirty-two channel micro direct memory access controller, data can be transferred to and from peripherals directly from memory, it can directly handle data transfer tasks intended for the processor, saving time, and alleviating the processor which frees space which can be devoted for raw data during data transfer requests.

Developers using the CC3200 can specify which sections of memory are retained, by multiples of 64KB, when forced to enter the low-power deep sleep mode covered later, losing efficient power usage for faster wake-up time. These options for memory retention encourages streamlined memory usage, and gives developers welcome flexibility in handling unexpected memory loss.

5.5.2.3.3 ROM

Read only memory stores high value data such as preprogrammed software like the operating system which is integral to start-up, interrupt and exception handling, and the very basic function of the device. This high value data should never be written into to because of its effect on the entire operation of the MCU, only read. Other sets of preprogrammed data usually stored in ROM also includes driver software to handle low level hardware operations, ensuring that the entire MCU runs smoothly and efficiently.

The internal zero-wait-state ROM is addressed at 0x0000 0000 to 0x0007 FFFF of the CC3200, which holds the Bootloader and peripheral driver library. The main function of the peripheral drivers are to save both SRAM and external serial flash space by performing initialization and control functions with bootloader capabilities while booting up or emerging from hibernation modes, with an option of interrupt driven or polled peripheral support. The Bootloader initially loads the program if serial flash memory is empty in the boot up process. Like ROM on all devices, all system sensitive code is prewritten to these addresses.

5.5.2.4 Direct Memory Access

The direct memory access controller of the CC3200 utilizes the architecture of the MCU to optimize memory usage, processor usage, and bus bandwidth by offloading data transfer tasks from the processor, and performing data transfers between memory and the peripherals. With thirty-two individually configurable channels, the micro direct memory access supports memory to memory, memory to peripheral, and peripheral to memory transfers, in modes that include simple transfers, pin-pong or continuous streaming transfers, and scatter-gather transfers for lists up to 256 transfers from a singer request, further increasing the efficiency of extended peripheral-MCU communication, while allowing a high level of flexibility to devote a variable number of resources to specific functions due to channel configuration.

Useful components of the micro-direct memory access design are channel dedication for built-in, on-chip modules, including bidirectional-two channels for receiving and transmitting between modules and the MCU. Further software capabilities allowing for greater control of the micro-direct memory access include designation of data priority levels zero for high and one for low, the availability of software directed channel requests, programmatically designated transfers sizes of logarithms of 2 up to 1024 bits, and can accept data types of sizes eight, sixteen, and thirty-two bits.

5.5.2.4.1 Scatter-Gather Memory Access

Scatter-Gather is a complex method of data transfer which is especially useful when data needing to be transferred is located at many disparate points in memory. To better comprehend the process of Scatter-Gather, while reading, look at **Figure 5.5.2.4.1-1**, below.

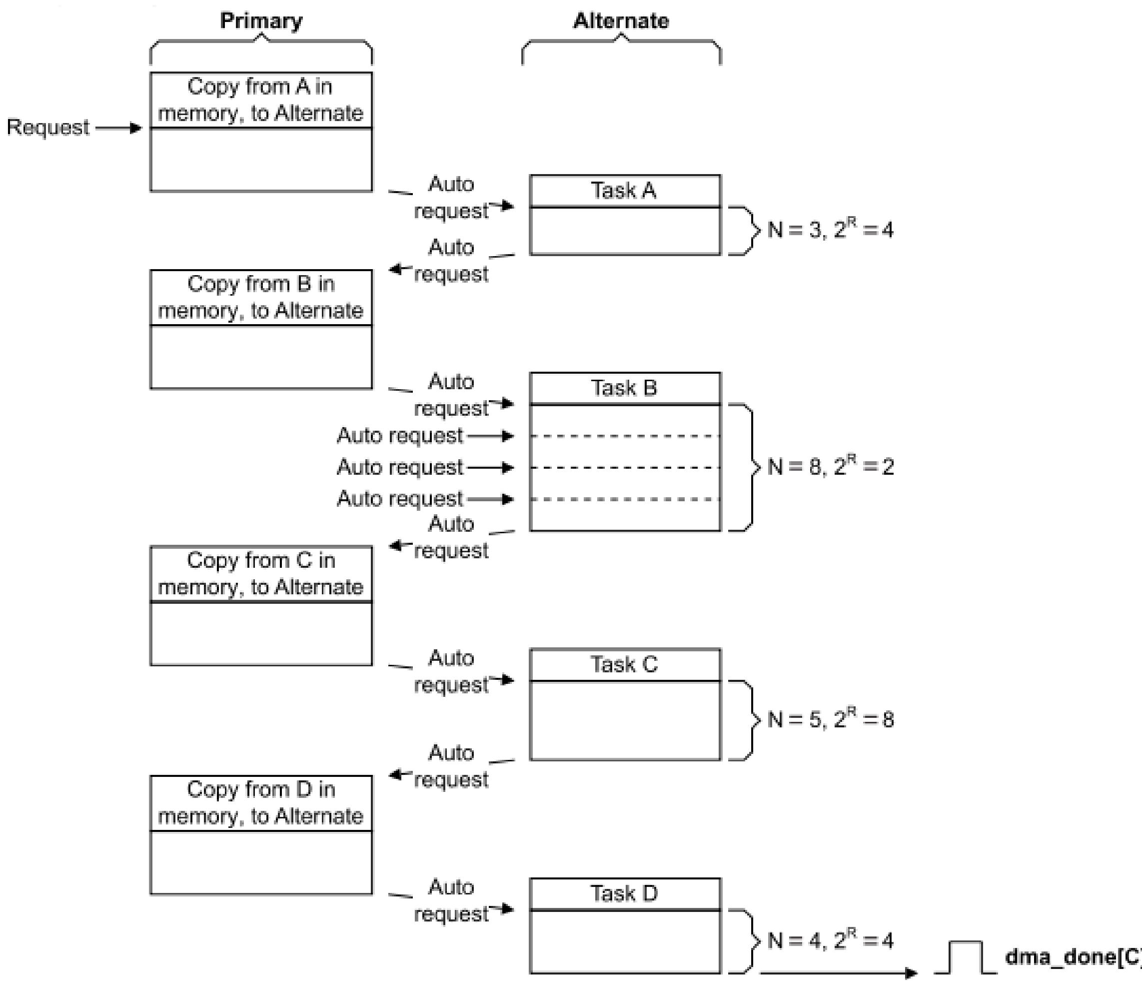


Figure 5.5.2.4.1-1: a visual representation of the Scatter-Gather data transfer algorithm
Reprinted with the permission of Texas Instruments

First the primary control structure programs the alternate control structure from a table in memory set up by the processor software. The table contains a list of control structures with the source and destinations end pointers as well as the control word of the desired data segments. Each table entry is then copied and executed in order in the alternate control structure while the micro-direct memory access controller focuses on transferring then executing each entry until the end of the list is reached, signaled by an interrupt. Lists can be looped back onto themselves if the final entry on the list contains the address of the list's beginning.

5.5.2.5 Power Usage

As the main focus of our project, energy efficiency is highly important as it is central to our overall design. The description of the CC3200 found on TI's website boasts of the MCU's high-functionality, low power design, claiming that is able to run four two years on the same amount of power supplied by two double-A batteries, but further analysis of the MCU is required to make a proper choice.

5.5.2.5.1 Power Supply Options

Two supply configurations are supported by the CC3200 for a supply between 2.1V to 3.6V, two 1.5V batteries or a regulated 3.3V direct power supply. In the case of brown-out or black out quick acting measures are initiated to ensure that data is retained so the device can resume once power levels re-stabilize.

5.5.2.5.2 Power Requirements: Low Power Modes

In the documents provided by Texas Instruments, the CC3200 has five operating states which requires differing minimum levels of current, hibernation, unconnected low-power deep sleep, active transmitting, active receiving, and idle-while-connected in low-power deep sleep. The terms active and low power deep sleep are two advanced low power modes referring to the state of the MCU. For our design, devices will always be in either active transmitting, active receiving, or connected low power deep sleep when in operation.

The active receiving and transmitting modes, meaning that the MCU is in the active state to either receive or transmit data, will be the two least commonly occurring states for our devices, and will need the most current to function properly, 59 mA for receiving and 229mA for transmitting respectively. Considering that the time to receive and transmit will only last a fraction of a second, these requirements should not make a serious dent in our overall power consumption.

The mode which our devices will be in for an overwhelming majority of the time will be the idle-while-connected in low-power deep sleep mode which will be when our MCU is not processing, receiving, or transmitting data, but listening for information from either our peripherals or the home's central database. In this mode, only the slow clock is active with the responsibility of listening and waking up the MCU to activate the fast clock to anticipate incoming transmissions. For this state the minimum current required to operate properly is 250 uA, 236 times lower than the next lowest low-power operating state, and is very impressive in comparison to the power usage of other MCUs.

5.5.2.6 Processor Details

The Arm Cortex-M4 is a high performance, low cost processor, needing a small pin count, low amount of power, and small amount of memory space to run efficiently and with low latency interrupt processing making it ideal for use with embedded devices. Its low latency is achieved with a nested vectored interrupt controller which fetches the interrupt and saves the processor state in parallel, tail-chains interrupts which prevents multiple unnecessary save states and restorations if multiple exceptions occur, and is closely integrated with the core processor, ensuring efficient late arrival interrupt processing. The wake up interrupt controller allows the processor to enter and reemerge from low power sleep states quickly while in time-sensitive scenarios. Three advanced high performance bus interfaces, the ICode, DCode, and system bus interfaces, allow for faster, efficient processing of different data types, and also features atomic read and write functions.

When debugging, the developer has access to all memory and registers, memory mapped devices, internal core registers when halted, and debug control registers. Also, the serial wire debug port and serial wire JTAG debug port allows developers to test the system's reactions to serial input. To ensure that code patches and breakpoints are possible to implement, the processor also features the flash patch and breakpoint unit.

5.5.3 Peripheral Sensors

Peripheral Sensors are needed to add the necessary functionality of a design to a microcontroller, by adding sensors and tools, microcontrollers are able to accomplish an incredibly large number of tasks with these function-specific customizable tools which use the microcontroller for central processing.

5.5.3.1 CC3100

The CC3100 has many capabilities similar to the CC3200, but is only available as a peripheral device and not a standalone MCU. Because the CC3100 does not have a processor some tasks such as hibernation control are built in to the device which would have otherwise been handled by the core. This peripheral would only be useful if paired with the MSP430 and 432, because they do not have built-in internet on a chip capabilities.

Like the CC3200, it is grouped with it as a first of its kind internet on a chip device, designated to assist in the design and communication of internet of things devices, and similarly, embedded systems, both of which have small data foot prints when compared to larger device, but also require some level of wireless communication to be fully operational. The CC3100 can connect to access points present in its environment or setup its own, ensuring that a stale internet connection is always present if there is power supplied to the device, a minimum requirement for all device-wide operations.

5.5.3.2 TMP007

A thermopile sensor, measuring the temperature of an object from a distance is exactly what is needed for processing the occupancy of a room with the above entry mounted sensor device. At wavelengths between 4 μm to 16 μm, the sensor is able to absorb infrared energy from an object within its field of view. With the internal reference temperature sensor, the math engine is able to measure the change in voltage across the sensor by calculating the temperature of the object. The TMP007 uses nonvolatile memory for storing variables for calibration.

Designed for portability and power optimization, it can easily fit in tight spaces while using standard surface-mount assembly processes to be placed against walls and other surfaces. Its low power consumption means it is suited for being supplied solely by battery-power which is exactly how the above wall mounted device will be powered. Offering a reduced feature set, The TMP006 has similar performance to its more expensive TMP007, but doesn't contain the math engine or nonvolatile memory which the TMP007 is assisted by. All calculations will be made by the processor. The sensor is able to operate from –40 degrees Celsius to 125 degrees Celsius, and is able to measure the object temperature beyond the device operating range if the device itself does not exceed this range.

5.5.3.3 TDC7200

A Time-to-Digital Converter, the TDC7200 uses for ultrasonic sensing for applications to measure water gas, and heat flow that it is perfect for the design. Our plan is to Pair the TDC7200 with the TDC1011, in order to measure the direction which people are moving so a proper room occupancy count can be taken, to determine whether someone is entering or leaving a room. The TDC7200 and TDC1011 can be a part of a complete Texas instruments set to measure this data.

Performing the same task as a simple stopwatch and the device measures the time between the start of the pulse, up to five times the pulse has been stopped. The ability to measure from start to multiple stopped pulses our team the ability to wait momentarily for another occupant to pass through, saving power by limiting the number of wireless communications are necessary. With this ability sensors could increment or decrement room occupancy by five instead of a total of one.

The TDC7200 has a self-calibrated time base which allows for temperature to drift without triggering the sensor to increment the occupancy count, this is very important to keep accurate measurements and is convenient for developers since this feature is already built-in. Self-calibration allows for accuracy in time to digital conversion to be precise up to a few picoseconds, making the TDC7200 suitable for our design and ensuring accurate counting of individual bodies which are only inches apart. To save power, the TDC7200 can be optimized by being placed in an autonomous multicycle averaging mode, which is beneficial for systems like ours running in battery power. The host microcontroller in this mode is able to hibernate while the TDC remains active and is only awoken when the TDC detects a break in the pulse.

With these many useful power saving, communication optimizing, and time measurement features, this device is perfect for working in tandem with the TDC1011 to track a physical body's direction of movement, passage through the entryway threshold, and the major task of tracking room occupancy, this device will be a suitable choice in our EnergyGuard system design.

5.5.3.4 TDC1011

Used for ultrasonic measurements of liquid and gasses, and for our above mounted sensor to track room occupancy, the TDC1011 is primarily used as a fluid identification or concentration, and proximity or distance measurement applications which is regularly found in industrial, medical, automotive and consumer industries. Usually coupled with the MSP430 or C2000 MCUs, which include wireless, power and source code, Texas instruments supplies an entire device for all aspects of ultrasonic sensing.

Texas instruments' at front end ultrasonic device offers programmability, robust features, diversity, and flexibility to assist developers pursuing a large variety of applicable functions and end product designs. The TDC1011 is organized to be able to handle multiple frequencies, transmission pulses, phases, and signal thresholds for use with a large variety of transduced frequencies and quality-factors. Similarly, the device allows for ultrasonic wave detection through a wide scope of distances and tank sizes because of the programmability of the receive path.

Selecting different modes of operation, the TDC1011 can be optimized for low power consumption, making it ideal for battery powered applications. The low noise amplifiers and comparators provide extremely low jitter, enabling picosecond resolution and accuracy. With two of these sensors installed with the above mounted sensor device, working together with the TDC 7200, physical bodies breaking the threshold of the entryway, the direction of entrance or movement, and the occupancy count between rooms and buildings can be accurately tracked.

5.5.4 Summary of Choice

After extensive analysis of the two microcontroller candidates to be used in our design, a decision must be reached. The CC3200 benefits from its already embedded internet on a chip hardware, ensuring that a crucial part is already present in this microcontroller design, its memory is streamlined to accommodate the needs of our design to perform simple communication tasks between the microcontroller and the database server, and the microcontroller and its peripherals. Also, the overall cost of the device is much lower because of its smaller, simpler memory, and its built-in internet on a chip capabilities, which must be purchased separately if using the MSP432.

Although the MSP432 has many benefits as well, with a large, well-organized, and leak-resistant memory, its flexible design, oriented to accommodate many peripheral devices, its large number of preprogrammed libraries to handle peripheral busses and internet MQTT protocol, and its low power requirements, it was cost which ultimately made us decide against the MSP432 in favor of the CC3200. The MSP432 design would be fully capable if not more capable to meet the requirements of our design, because of its large memory, many useful libraries, and multi-peripheral design, but cost is a very serious concern for any project. And although the CC3200 is less capable than the MSP432 to handle the tasks of our design, it is still possible to use the CC3200 to accomplish our goals with some efficient well optimized code and hardware design from our team.

The ultimate questions which were fundamental in our decision making process were can this piece of hardware handle the necessary functions of our design, and what will the effects of this choice be on our project budget. And not only from the simple viewpoint of maintaining budgetary stability, but to also meet the requirements of ensuring our design goals are met in the most cost-efficient way, our team can be ever closer in engineering a solution which is competitively priced when compared to similar designs currently on the market.

5.6 Relays and Relay Driver

A relay is a switch that is operated electrically. There are relays that can handle the high amounts of power to control a motor, known as a contactor. Then, there is the type of relay that we will be using that controls power circuits, known as a solid-state relay (SSR). SSRs work by detecting a small external voltage. When this voltage is detected, the SSR will make an open circuit and not allow for current to pass through to the output. When this voltage is not detected current can pass through the closed circuit freely. The SSR consists of three main parts: a sensor that responds to a voltage input (control signal), a solid-state electronic switching device the switches the power, AC or DC, directed toward the load circuit, and a coupler to allow for the control signal to activate the switch with no mechanical parts. SSRs use semiconductors, such as thyristors and transistors, to switch high value currents of up to 100 amperes. SSRs have many advantages, such as very fast switching speeds and no physical components so there is no wear. There are also disadvantages though, was with every device. Some disadvantages include the inability to withstand momentary overload, higher on state resistance, and limited switching arrangements. We will make sure that overload does not occur by using a surge protector; we will choose a SSR with lower on state resistance, so that high amounts of power are not dissipated by the resistor; and, the limited switching arrangements is not a problem as we only need the switch to open and close.

5.6.1 Relay

There were three relays we chose from they include the S108T02 Series Solid State Relay, the CPC1998 AC Power Switch, and the CPC 1965Y AC Solid State Relay. The three will be described below.

<u>S108T02 Series Solid State Relay</u>
The S108T02 Series Solid State Relay is made up of an infrared emitting diode (IRED), a Phototriac Detecter, and a main output Triac. The S108T02 is suited to control high voltage loads. The functional diagram is shown in **Figure 5.6.1-1**. Its features include:

- Output current less than or equal to 8 A
- Zero crossing functionality
- High peak off-state voltage of 400 V
- 3 kV isolation voltage from input to output
- Lead-free terminal option
- Screw hole for heat sink

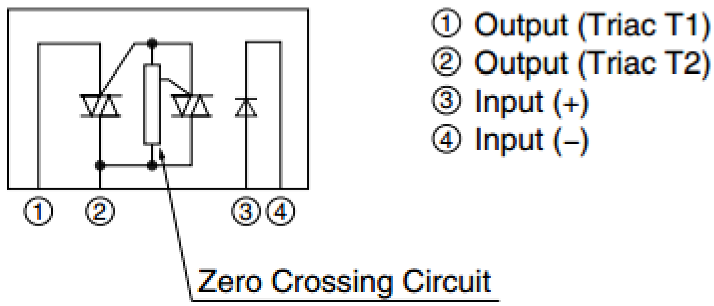


Figure 5.6.1-1: Functional Block Diagram for the S108T02
Reprinted with permission from Sharp Microelectronics

<u>CPC1998 AC Power Switch</u>
The CPC 1998 AC Power Switch is an AC Solid State Switch. The functional block diagram and pin layout is shown in **Figure 5.6.1-2**. Its features include:

- High load current
  - Up to 20 A with a 5 ºC/W Heat Sink
  - Up to 5 A without a Heat Sing
- 800 V_p Blocking Voltage
- 5 mA Control Current
- Zero crossing functionality
- Ceramic Pad for Heat Sink
- 2.5 kV isolation voltage from input to output
- DC control, AC output
- Optically isolated
- High noise immunity

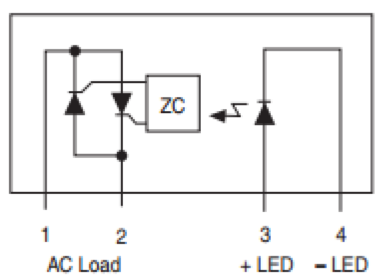


Figure 5.6.1-2: Functional Block Diagram for the CPC1998
Reprinted pending permission from IXYS

<u>CPC1965Y AC Solid State Relay</u>
The CPC 1965Y is an AC SSR that uses patented waveguide coupling with dual power silicon controlled rectifier (SCR) outputs. This is an alternative to Triac circuits. Zero-cross circuitry is include, just like the other two relays to keep the AC load from producing transients. The high isolation voltages that are included in all three relays is beneficial because it prevents electromagnetic interference. The functional diagram and pin layout is shown in **Figure 5.6.1-3**. Its features include:

- Load current up to 1 A
- 600 V_p Blocking Voltage

- 5 mA sensitivity
- Zero crossing detection
- 3.75 kV isolation voltage from input to output
- DC control, AC output
- Optically isolated
- High noise immunity

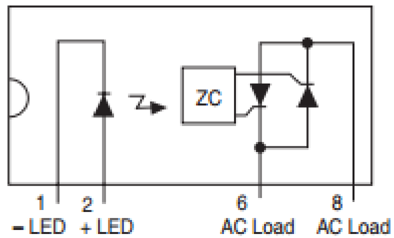


Figure 5.6.1-3: Functional Block Diagram for the CPC1965Y
Reprinted pending permission from IXYS

We chose the S108T02 Series Solid State Relay for our circuit. It had a very desirable max current and voltage output. It also was small, which was desirable, considering we will need three of them for our design. There wasn't too much more we were concerned with when it came to the desired function and specs of the SSR.

The SSR is in the ON state when the input signal current (I_F) is between 16 – 24 mA. The SSR is in the OFF state when the input signal current is between 0 – 0.1 mA. In the ON state, the output Triac is a closed circuit. In the OFF state, the Triac is an open circuit. A problem may occur if the voltage across the Triac increases at a rate faster than 30 V/µs. If this occurs, the Triac may switch to the ON state. This problem is fixed by incorporating an RC snubber circuit. The resistor and capacitor will be set at 47 Ω and 0.022 µF, respectively. During testing we will see if there is undesirable switching occurs. If it does, we will adjust the resistor and capacitor values accordingly.

5.6.2 Relay Driver

There were three relay drivers we chose from they include the ULN2003LV 7-Channel Relay and Inductive Load Sink Driver, the DRV120 Power-Saving Current Controlled Solenoid Driver, and TPL9202 8-Channel Relay Driver with Integrated 5-V LDO and Brown-Out Detection. The three will be described below.

ULN2003LV 7-Channel Relay and Inductive Load Sink Driver

The ULN2003LV is a low-voltage and low-power 7-channel Darlington transistor array. There are seven low output impedance drivers. These drivers support low voltage relay and inductive coil applications. These low impedance drivers also minimize on-chip power dissipation. The functional diagram and pin layout is shown in **Figure 5.9.2-1** and each channels functional block diagram is shown in **Figure 5.6.2-2**. Its features include:

- 7-Channel High Current Sink Drivers

- Supports up to 8V Output Pullup Voltage
- Supports a Wide Range of 3V-to-5V Relay and Inductive Coils
- Low Output VOL of 0.4V with:
  - 100mA Current Sink per Channel at 3.3V Input
  - 140mA Current Sink per Channel at 5.0V Input
- Compatible to 3.3V and 5.0V Microcontrollers and Logic Interface
- Internal Free-Wheeling Diodes for Inductive Kick-back Protection
- Input Pulldown Resistors Allow for 3-stating the Input Driver
- Input RC-Snubber Eliminates Spurious Operation in Noisy Environment
- Low Input and Output Leakage Currents
- Easy to use Parallel Interface
- ESD Protection Exceeds JESD 22
  - 2kV HBM, 500V CDM
- Available in 16-Pin SOIC and TSSOP Packages.

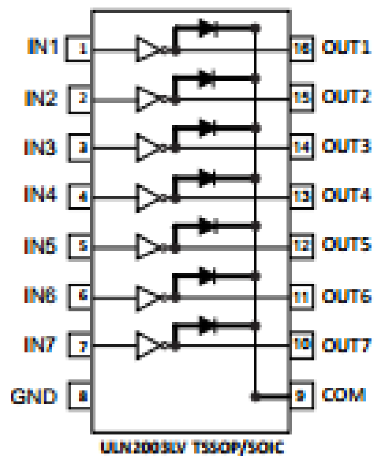


Figure 5.6.2-1: Functional Block Diagram of the ULN2003LV

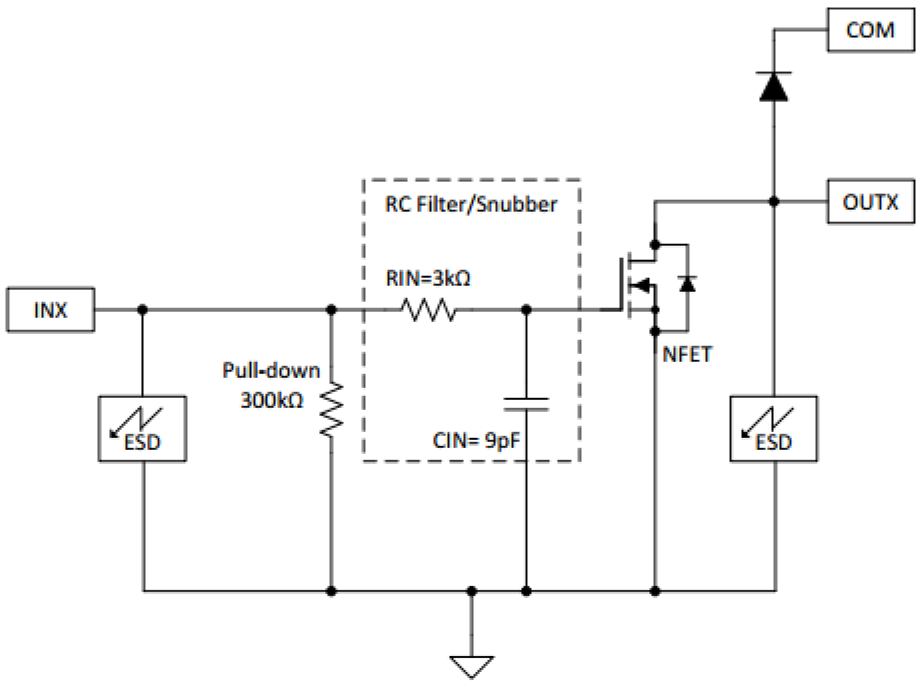


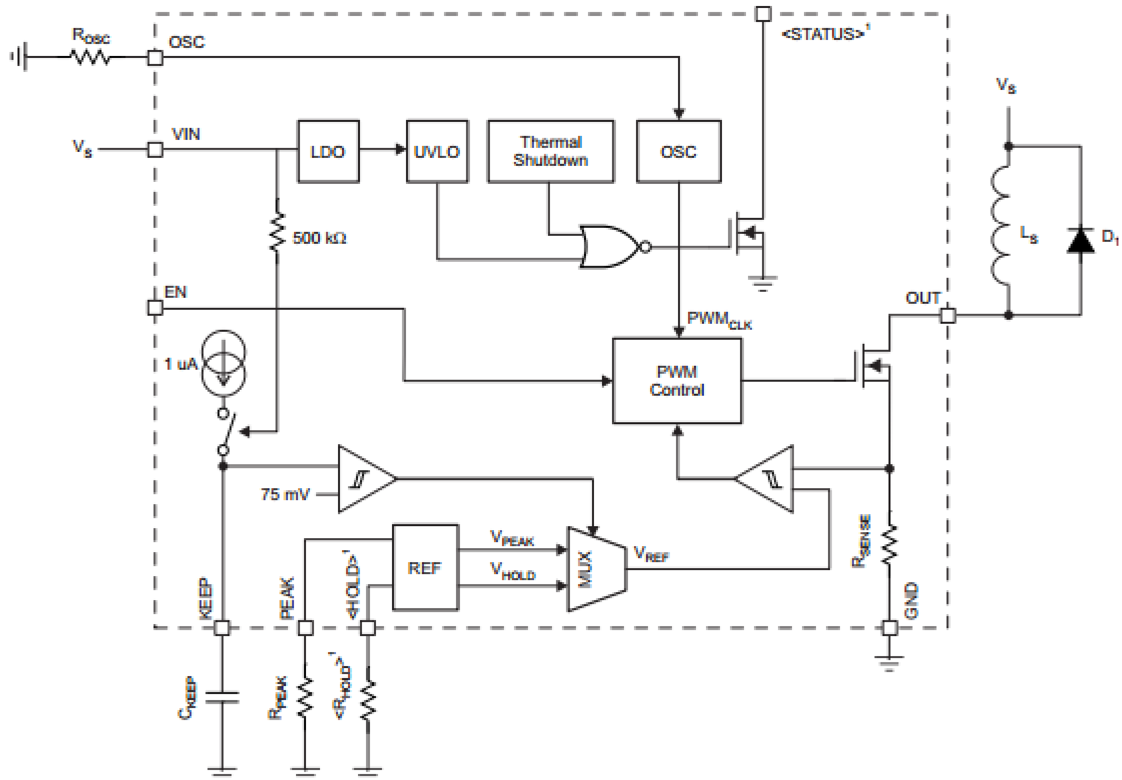
Figure 5.6.2-2: Functional Block Diagram for the Channels of the ULN2003LV

DRV120 Power-Saving Current Controlled Solenoid Driver

The DRV120 is a pulse width modulation (PWM) current driver for solenoids. It is designed to regulate the current with a waveform to allow for activation while also reducing power dissipation within the device. The solenoid current is ramped up very quickly to make sure that the relay opens. After the ramping, the current levels off at the peak value to ensure correct operation, then it drops to a hold current to avoid thermal problems and reduce power dissipation. The duration of the peak current is set using an external capacitor. The peak current, as well as the hold current and PWM frequency values are set using external resistors. The functional diagram is shown in **Figure 5.6.2-3**. Its features include:

- Integrated MOSFET with PWM to Control Solenoid Current
  - Integrated Sense Resistor for Regulating Solenoid Current
- Fast Ramp-Up of Solenoid Current to Guarantee Activation
- Solenoid Current is Reduced in Hold Mode for Lower Power and Thermal Dissipation
- Peak Current, Keep Time at Peak Current, Hold Current, and PWM Clock Frequency can be Set Externally. They can also be Operated at Nominal Values without External Components.
- Internal Supply Voltage Regulation
  - Up to 28V External Supply
- Protection:
  - Thermal Shutdown
  - Under voltage Lockout (UVLO)
  - Maximum Ramp Time
  - Optional STATUS Output
- Operating Temperature Range: -40^{o}C to 105^{o}C

- 8-Pin and 14-Pin TSSOP Package Options



¹Available only in the 14-pin package

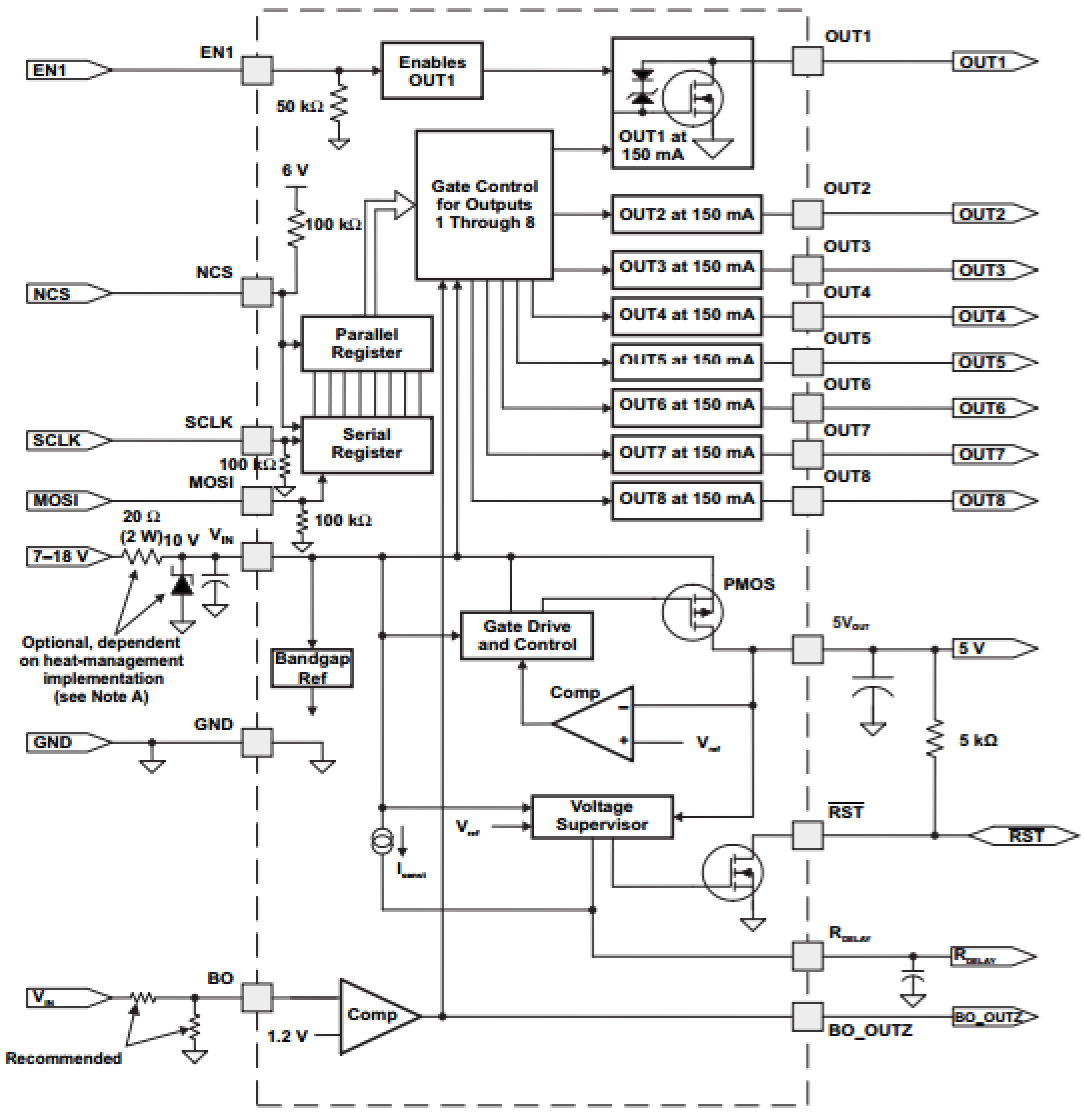
Figure 5.6.2-3: Functional Block Diagram for the DRV120

<u>TPL9202 8-Channel Relay Driver with Integrated 5-V LDO and Brown-Out Detection</u>

The power supply of the TPL9202 provides regulated 5V output which powers the system microcontroller and drive the eight low-side switches. The brown-out detection output (BO_OUTZ) sends the system a warning about a drop in supply voltage, allowing for the prevention of potentially hazardous situations. A serial communications interface controls the eight outputs. Each of the outputs is connected to an internal snubber circuit, which is used to absorb the inductive load when the driver is in the OFF-state. The functional diagram is shown in **Figure 5.6.2-4**. Its features include:

- Eight Low-Side Drivers with Internal Clamp for Inductive Loads and Current Limiting for Self-Protection
  - Seven Outputs Rated at 150mA and Controlled through Serial Interface
  - One Output Rated at 150mA and Controlled through Serial Interface and Dedicated Enable Pin
- 5V ± 5% Regulated Power Supply with 200mA Load Capability at V_{IN} Max of 18V
- Internal Voltage Supervisory for Regulated Output
- Serial Communications for Control of Eight Low-Side Drivers

- Enable/Disable Input for OUT1
- 5V or 3.3V I/O Tolerant for Interface to Microcontroller
- Programmable Power-On Reset Delay Before RST Asserted High. Once 5V is within Specified Range
- Programmable Deglitch Timer before RST Asserted Low
- Programmable Brown-Out Feature
- Thermal Shutoff for Self-Protection



A. The resistor and Zener diode are required if there is insufficient thermal-management allocation.

Figure 5.6.2-4: Functional Block Diagram for the TPL9202

We chose the ULN2003LV 7-Channel Relay and Inductive Load Sink Driver because it can control multiple different relays using multiple different inputs. It can drive low-voltage relays, LEDs or resistive loads. The ULN2003LV contains an input RC snubber circuit that prevents the Relay from switching randomly in noisy environments. Further enhancement of this noise tolerance can be achieved by connecting a 1kΩ to 5kΩ resistor in series with the input. A 300kΩ input pulldown resistor is used to allow the input drivers to be tri-stated. When a high impedance driver is connected to an input, the input is detected as a low-level input and remains in the OFF-state. Another feature of the ULN2003LV is the flexibility to increase the current sink capability by connecting adjacent channels in parallel. The seven channels of the ULN2003LV can all be connected in parallel, allowing for support up to 1A of load current.

5.7 Socket Types

According to the International Electrotechnichal commission (IEC) the two socket types used in the United States are socket types A and B. Each type must be rated for 120 V at 60 Hz for the power in the United States. Our project will require type B since our design needs an Earth or ground. The IEC Standard TR 60083:2015 outlines the socket types of that will be used for domestic use not only in the United States but worldwide. In order to comply with this standard any domestic use system will need have an AC input no less than 50 V but no greater than 440 V. If we do not use type A or B then we will not comply with the standard and our product will not work and could cause damage. Therefore it is important we comply with the standard and use the correct outlets for our country and voltage range.

5.7.1 Connectors

Our project Specifications require 3 female connectors for our power strip and at least 1 male connector for the power supply of the power strip. The IEC Standard 60320 Appliance couplers for household and similar general purposes applies to the sockets we will be using. The IEC standard outlines that connectors should be rated for no more than 250V and a max current of 16 A. The connector types we have chosen for our design are C-13 and C-14. **Figure 5.7.1-1** shows a diagram of what the connector looks like and the specs. The Underwriters Laboratories (UL) is a safety consulting and certification body. Mostly all electronics in the US have a UL rating. Our connector has been rated by UL for a max of 15A. A range of 10 – 15 A should be more than enough to satisfy our project requirements.

C-13 Female C-14 Male Ratings: 250 V 10 A IEC 15 A UL	

Figure 5.7.1-1 C13 Connector

The part number for the actual connectors we will use in our design are Bulgin PX0675/PC for Female connector and Bulgin PX0580/PC for our male connector. Both parts comply with UL and IEC standards for safety. The female PX0675 connector is useful for us since it has three connection pins. One live pin, one earth pin, and a neutral pin. We will need these pins so we can receive power readings and control power to each of the 3 connectors. We also need the 3 pins to comply with the IEC socket standard mentioned earlier. These will be Type B sockets since the connector will have a ground. Each of the C-13 sockets will be IEC compliant with a voltage of 120 V and 60 Hz frequency.

5.7.2 Connectivity

Each of the 3 C-13 connectors will be coupled with a relay. The relays will switch the power to the connectors on or off depending on the signal received from the relay driver. Please refer to **Figure 5.7.2-1** for a high level diagram for the connectors. The functions of the relay and the relay driver are mentioned in the relays section. This diagram shows that the "Live" line from the power supply will energize the C-13 connectors with 120 V to act as a common domestic socket. Using Line_A as an example, the relay driver will send a 3.3 V signal to relay A and the relay will switch to a non-energized line. This will effectively turn off the C-13 A connector. All 3 connectors and relays will behave like this.

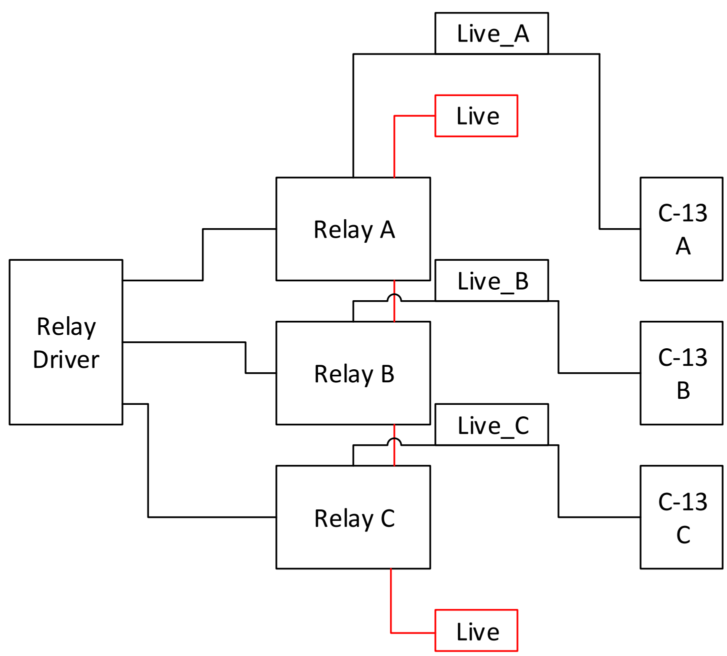


Figure 5.7.2-1 Relay Coupled Connectors

5.7.3 Power Supply

The power will be supplied from a wall outlet meant for domestic or home use. Mentioned earlier this means that our supply for the United States will be an AC 120 V at 60 Hz. Most of the components such as the relays are expecting a DC input. It is required that we add an AC to DC converter. We will do this on the input of the power supply. **Figure 5.7.3-1** shows the circuit we will use for the conversion. The circuit uses the AC/DC converter made by CUI Inc. The part number is VSK-S2-5U.The converter has two inputs and two outputs. The input voltage for the part has a max of 305 Vac. Our input will be max 120 Vac so the part can handle the input without damage. The output voltage will be 5 V. The 5 V output will be sufficient to power the relays and relay driver plus any other components in the design. The component on the live input is a fuse part number 6125FA1A. The fuse is rated for 125V and will be used for protection. So if any surges happen beyond our 120 V power supply, the fuse will activate and protect the circuit. The Protection Diode (PD) is put on the input and output of the converter. The protection diode will protect the input and output from any surges that may happen. The converter is followed by a linear regulator. This regulator will input 5 V and output 3.3 V. This is needed since some elements of the design such as the LED's need a 3.3 V signal. This signal will be called VCC in our circuit diagrams. The VCC line will routed through a jumper header and supply the correct voltage for any component that needs the VCC line and 5 V is too great.

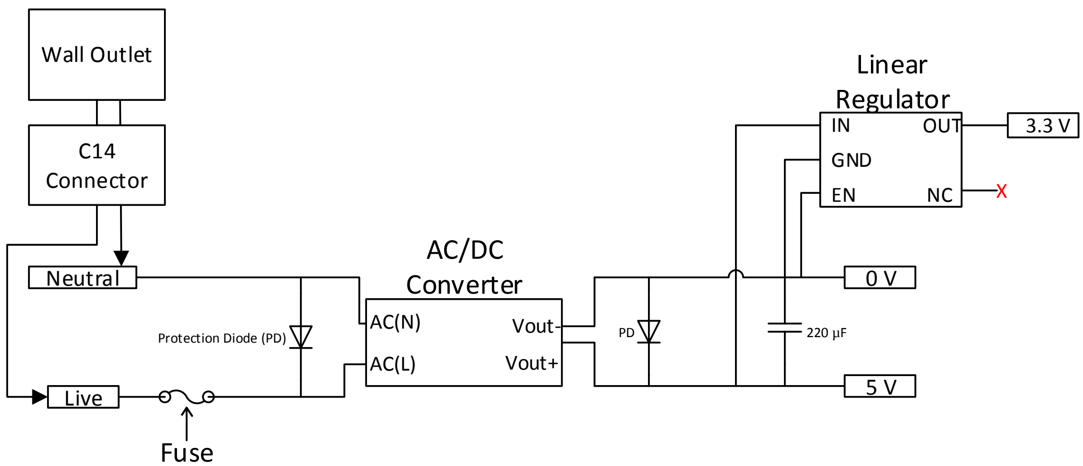


Figure 5.7.3-1: AC/DC Conversion and Linear Regulator

5.8 LED Design

The design will have a LED network. Each LED will illuminate according to conditions that we set up in the software. **Figure 5.8-1** shows what the LED network will look like. This figure shows 7 LED's but we may need more or less. The Circuit itself is very simple. Just like any LED a resistor is needed so the current from the power supply will not destroy the LED. The Resistor Value is chosen depending on the Power supplied to the LED network and the forward bias voltage of the LED. Using a simple KVL formula, the resistor value is chosen by the equation $R = \frac{V_S - FB}{i_D}$ where V_S is the source voltage, FB is the forward bias voltage, and i_D is the diode current. The values for FB and i_D can be obtained from the datasheets.

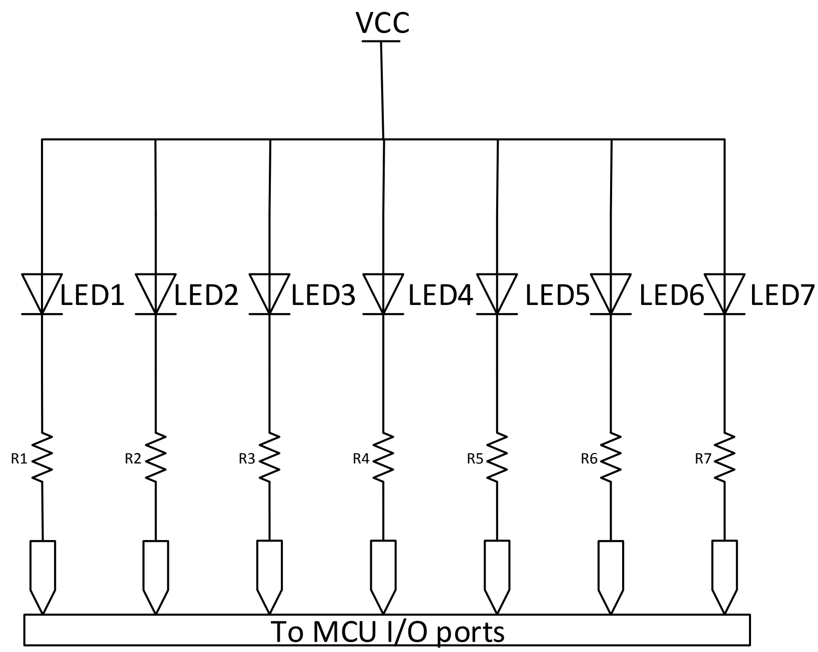


Figure 0-2: LED Design

5.8.1 Diode Function and Color

We would like the LED to network to inform the user for the following Reasons:

- Normal Operation
- Sockets LED
- Sending/Receiving Data
- Problem LED
- Service Mode

The Normal Operation LED ideally will always be on. This will tell the user that power is on and everything is working correctly with the device. The “Sockets” LED will tell the user if one of their sockets is currently on or off. There will be one for each socket connection on the power strip. A LED will blink when Sending or Receiving data from the database using the wireless signal. The problem LED will tell the user if the device has stopped functioning in some way. The service mode LED will be on if the device is being serviced somehow. For example, updating MCU software would require a service mode.

The color of the LED is very important. It would be ideal to have a different color for each function mentioned. The fact is that each LED color has a different Forward bias voltage. The design can get complex and confusing for so many colors. To keep the design as simple as possible, we have chosen to use LED’s that have a forward bias voltage of 3.3 V. The colors we have chosen with this Forward Voltage are Blue and White. The software will control the logic of the LED network. Since we only have two colors position of the actual LED will be important. For example, the socket LEDs need to be close to each socket and possibly some small lettering on the PCB or casing to indicate the LED function.

5.9 Auxiliary Schematics

Our system has some auxiliary components that are important to the functionality although they are not shown in the discussion of the main systems. Here is where the discussion of any components that needed such as pin headers and the flow of the circuit. Pin headers can be used to connect ribbon cables for debugging or other functions. Also some pins can be used so that jumpers can be applied. This design uses both cases. **Figure 5.9-1** shows the pin headers that will be used as jumpers and their connections. The 4 pin header creates the 3.3 Vcc line that is needed anywhere a component needs a 3.3 V input. The 3 pin header is used on each of the 3 relays. This header will allow the flow to the coil so the relay will turn on or off. Although these headers seem like insignificant our system would not work unless the jumpers are correctly installed on the pins.

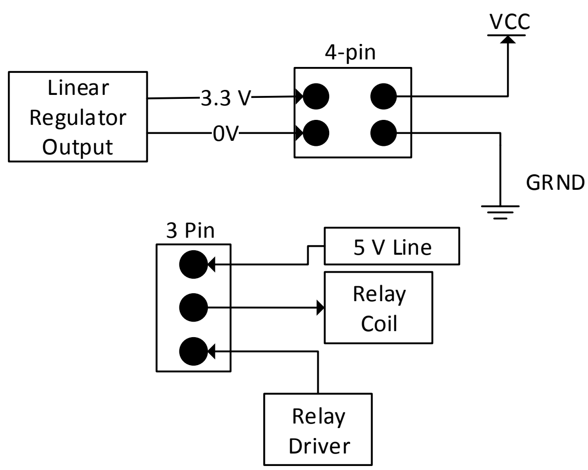


Figure 5.9-1: Headers

It is possible for us to use a header for debugging the MCU once the board is assembled. This circuit found in **Figure 5.9-2** uses a 14 pin male header for a ribbon connector. The ribbon connector will run to a debugger and then to a PC. This will allow us to make changes to our code once the PCB is complete. This can be useful since it is possible that mistakes could be found and need to be changed after completion of the design.

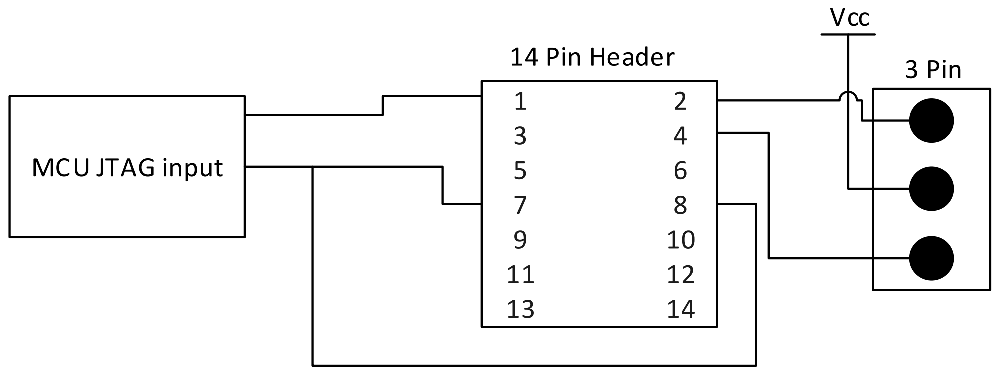


Figure 5.9-2: Debugging interface

5.10 Testing

Our project has 3 main devices. The power strip, the mounted device, and the HVAC and Dryer control system. Each of the 3 devices need to be tested in order to make sure all components are functioning and the system meets the design specs. Also each device needs to be able to communicate with the server so that data can be collected and the system can act on the data collected. So each device will also need to be tested for data functionality.

5.10.1 Power Strip

In order to test the power strip we have to test each sub-system of the power strip separately. The sub systems to test include:

- Voltage Measurement
- Current Measurement
- Relays and Relay Driver
- LED function and logic
- Power Supply
- Surge protection

Testing the Measurement circuits would mean to put in known and controlled voltage to the input. We will use a power supply with a constant AC voltage that is known. We will then make sure the MCU can read the Voltage and Current and display it. This will also test our software development for this part of our system. The Relays and Relay driver testing will include testing for the 3 sockets. We will need a 120 V 60 Hz line to power the sockets. We also need a 5 V line to send to the relay to switch. Each relay will be tested by sending signals from the relay driver. If the testing works correctly, then the relay should switch from the 120 V live line to a line not connected to anything. This procedure will test the Relay Driver, the relays themselves, each socket, and the software used for the driver logic.

There are two parts to testing the LED's. First each LED must be tested to see if it will illuminate and then the MCU logic needs to work. Testing the function of the

LED is the simple part. Each LED need to be connected to its proper forward voltage and in series with a correct resistor amount. Apply a source voltage and the LED should illuminate. Each LED must be tested separately before connected to the system. This so we know that it is not a faulty LED. The LED Logic must be tested by loading the software onto the MCU that will control the LED logic functions. Each LED environment such as normal operation will have to be created so that we can see the software logic is correct.

The power supply testing will include testing of the C-14 connector, the AC/DC converter, and the linear regulator. We will need a 120 V 60 Hz power supply for the circuit testing. Using an oscilloscope, we should see a 120 V amplitude wave on the input of the converter. The output of the converter should yield a DC voltage with no wave. The voltage of the DC output will vary with the converter type. For our design, we want to see a 5 V output. The linear regulator needs to be tested in the same fashion as the converter. We can use just a DMM here since both the input and output will be DC voltage. A DMM should measure 5 V input and 3.3 V output.

Lastly the Surge protection should be tested. Each of our main sub-systems has some sort of protection element. Mainly our system has electrostatic discharge (ESD) diodes, Metal oxide varistors (MOVS), and Fuses. Testing this is a challenge since it is not practical to hit our circuit with a very large voltage. This is dangerous and if our circuit is not correct then our components will be damaged. The only way to test is make sure the MOV's, ESD's and Fuses meet the specified manufacturer's requirements. The datasheets for a MOV give test specifications that must be met for it to work. Making sure the components meets these conditions will determine that the MOV works correctly. The datasheets for the ESD's list all of the electrical specifications including the types of waveforms to test the components. Each ESD will be tested in this manner. Any fuses can be visually inspected to see if it has blown. If the housing is not transparent, then use an ohmmeter to test the resistance of the fuse. The fuse should give a resistance value.

5.10.2 Mounted Device

Each component of the device needs to be tested separately. The device will consist of at least 3 sensors which need to be tested according to the specifications found in the datasheet for the device. The types of sensors we plan to use are:

- Motion Sensor
- Sensor 2
- Sensor 3

Each sensor will need to be tested for functionality. The datasheets will provide the benchmarks we will test for each sensor. The motion sensor we will most likely use is passive infrared sensor or PIR. This sensor uses infrared light to capture motion in its field of view. According to the datasheets a simple way to test the motion sensor is connect it to an LED using the following circuit diagram of **Figure 5.10-**

1. The resistance or R will depend on the type of LED that is used. The idea of the circuit is that the LED is connected to the digital out of the PIR sensor. This output will give a digital state depending on if the PIR detects anything in its view. So after the PIR has stabilized then set the sensor off and the LED should light up. This will test to make sure that the PIR is functioning as it should. Then the MCU wireless connection will need to be tested to make sure the data from the sensors is recorded properly and the system reacts as expected.

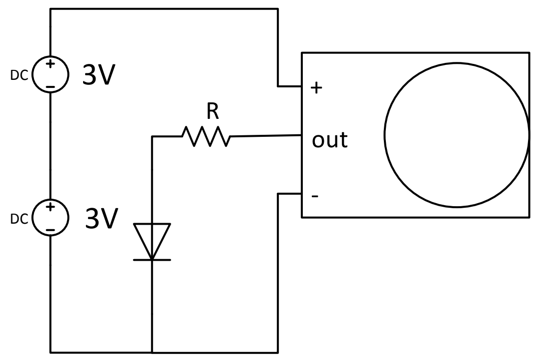


Figure 5.10.2-1: Testing PIR Sensor

5.10.3 HVAC and Dryer

This system will consist of the following:

- Thermostat to control HVAC
- LED Network for HVAC
- Dryer represented by a LED
- Wireless MCU

The thermostat system will consist of four Seven Segment displays to show the current temperature. We need to make sure that the numbers will change according the temperature that is passed into the system. Each of the four seven segment displays would have to be tested to prove that they can display the correct number ranges 00-99. If we use a pentiometer, we have to make sure that the correct resistance is attained at each point on the wheel. The wheel will turn from the least resistance to the highest resistance. We would make sure that the numbers on the display change with respect to the resistance on each point of the wheel. The push buttons would be easier to test since all we would have to do is hold the push button down and watch the numbers increase or decrease according to how we wire the circuit.

Also, we need to test that the proper signals will be sent to the MCU in order to turn the HVAC on or off according to the temperature. We will have a red and blue LED that will be used to represent when the HVAC is cooling or heating. The Cooling and Heating function will have to be tested. Each LED must light up when the temperature of the system drops below or climbs higher than the setpoint temperature. The dryer will be represented by a LED controlled by the MCU. The MCU software will control when the dryer should be on or off. All of the logic will need to be tested. The Hardware for this device is relatively simple. The most testing will be on the software code and logic.

5.10.4 Software Development

Our project is going to have a large amount of software to upload to the microcontroller in order for the system to function how we would like. Also the database will house all of the data the system creates. The items we need to test are:

- Database and Server
- Android App GUI

The database will be housed on the server and will collect all of the data the system creates. The data is sent to server via the microcontroller's wireless connection. We need to test and make sure any power data is being sent and housed correctly in the correct tables. The best way to do this is to only send one signal at a time. For example, we can create a voltage and a current on one of the microcontroller's inputs. That data should be routed to the correct location. Check the location in the database and make sure the data is populated. The android app GUI is going to take extensive and tedious testing to make sure every aspect is working correctly. We should test it on as many device as we can to ensure there is no problem with cross device implementation. A good way to test this system is see that it can handle every feature that it is programmed to accomplish. Then try to create errors that could crash the app and make sure error handling works correctly. This is a long and tedious process but will mitigate any issues that may come up a later time.

5.10.5 CC3200 Microcontroller

The microcontroller will have to go through very extensive testing. Our prototypes will be tested with a Launchpad to make sure all of our circuits work. The Launchpad will allow us to make quick adjustments on the fly while perfecting our design. Once we have something that works correctly we will load a standalone microcontroller and put that one into the circuit. Testing the microcontroller means we have to test and make sure every input and output is accounted for and working as we have designed it. This will require testing each input individually and making sure the functionality is correct. At the same time our circuits such as the voltage reading will be tested for functionality. Each software block will be reviewed at the same time as testing each input. This is to make sure the code for that microcontroller input is performing the correct functions. Our system will implement a header that will connect to a ribbon and we will be able to access the microcontroller's code through a PC and USB debugger. Extensive testing of this function is imperative so that we can make sure to correct anything that may go wrong with our microcontroller software. The only way to test this is try out the connection and see if it works.

5.10.5 Overall Function

Our design has a lot of functions that need attention. We need to test the input and output of each component either with a DMM or oscilloscope. For example, make sure everywhere the Vcc line is used that the voltage is in fact 3.3 V there. Having this incorrect may damage components. Each voltage and current will need to be manually read on an instrument and then make sure that matches with what our system. Each header pin will need to be measured to make sure that the pin of the header is what we expected. Every line that will supply power to a component will need to be tested for reliability. If any of these voltages are off then a component could be damaged.

6 Above Entrance Mounted Device

A feature that ties into our smart power strip is the above entrance mounted device. This device includes a number of sensors that monitor the inhabitants of a room. This feature will help us manage power passively and save power automatically. This section explains its functionality and how we will implement the device into our product.

6.1 Functionality

A number of sensors will be used on this device. These sensors communicate to the MCU and in turn to the database. The sensors are used to keep track of the inhabitants in a room. As people go into the room, the sensors will send data to the MCU, which will send data to the database, telling it to increment the room count by one. As people leave the room, the sensor will send data to the MCU, which will send data to the database, telling it to decrement the room count by one. The user can update their preferences in the web app using their android smartphone. The user will be able to control which sockets it would like off when the number of people in the room is recorded as zero. Any socket that is set to be on will always be on, no matter how many inhabitants are in the room. In order to make sure that the sensors function properly, we will ask the user to identify that there is nobody in the room during initial setup of the sensor.

6.2 Data (Diagrams Included)

This section will describe and explain the flow of the data in regards to the Above Entrance Mounted Device (AEMD). The main objective of this device will be to sense a change in the occupancy count of the room. The objective will not be to keep track of the occupancy count since this will be done by the database. However, the objective will be either to increment or decrement the value that is stored in the database. The flow of data in regards to the AEMD, as modeled in **Figure 6.2-1,** will only begin at the device because the device will not need to know any of the surrounding data. The database, and the web application will only need to know about the data that results from the AEMD. Therefore, the data flow in regards to the AEMD will begin at the sensors that are attached to the microcontroller on the AEMD. Once the sensor senses a change or the sensor is triggered, the signal will be sent to the microcontroller on the AEMD.

The data flow will then progress from the sensors to the microcontroller where the microcontroller will perform calculations on the signal, in order to understand whether this was an increase or a decrease in the occupancy count of the room. After the microcontroller knows whether to increment or decrement the occupancy count of the room, this data will be sent from the microcontroller to the web server. The web server will then pull the occupancy count that is already stored in the database, and either increment or decrement the value. Once the value is updated, the server will then store the value in the database again. This value will be used throughout the web application, and the Power Strip, as the functionalities will differ based on the value of the occupancy count of the room.

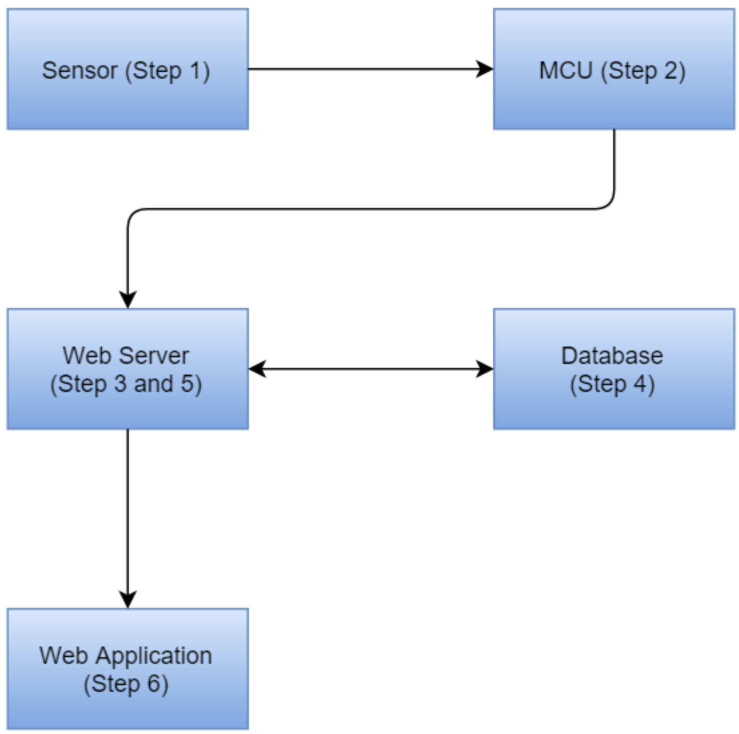


Figure 6.2-1: Data Flow Starting at AEMD

7 HVAC/Dryer

One aspect of our project is the communication between the HVAC and dryer. We are not designing an HVAC or dryer for commercial use, per se. The reason for our design is for the ability of power companies to distribute lower amounts of power to a home at a time, without general spikes in necessary power distribution. This will be done by only letting the dryer be on when the HVAC system is off. The reason we chose the HVAC and the dryer as the two appliances we do not want to be on at the same time is because they are the two appliances that consume the most power, as shown in **Table 7-1** below.

Table 7-1: Average Power Consumption of Common Household Appliances

Appliance	**Average Power Consumption (Watts)**
HVAC	5,000
Dryer	3,400
Oven	3,000
Hair Dryer	1,500
Dishwasher	1,500
Coffee Machine	1,500
Microwave	1,500
Toaster	1,100

Our design will consist of a very basic smart thermostat and an LED that represents if the dryer is on or off. The thermostat will have a Wi-Fi-enabled microcontroller, so that we can interact with the thermostat from a smartphone. The thermostat will have two 2-digit 7-segment displays; two displays will be used to represent actual temperature and two displays will be used to represent desired temperature. The desired temperature can be changed either manually or over Wi-Fi, using a smartphone. Our method for manually changing the temperature is by using a potentiometer to recognize the turning of the thermostat housing (similar to the Nest thermostat). Finally, there will be a push button to change between heating and cooling, and there will be a switch to turn the thermostat on and off completely and another switch to turn the dryer on or off. The block diagram is shown in **Figure 7-2** below.

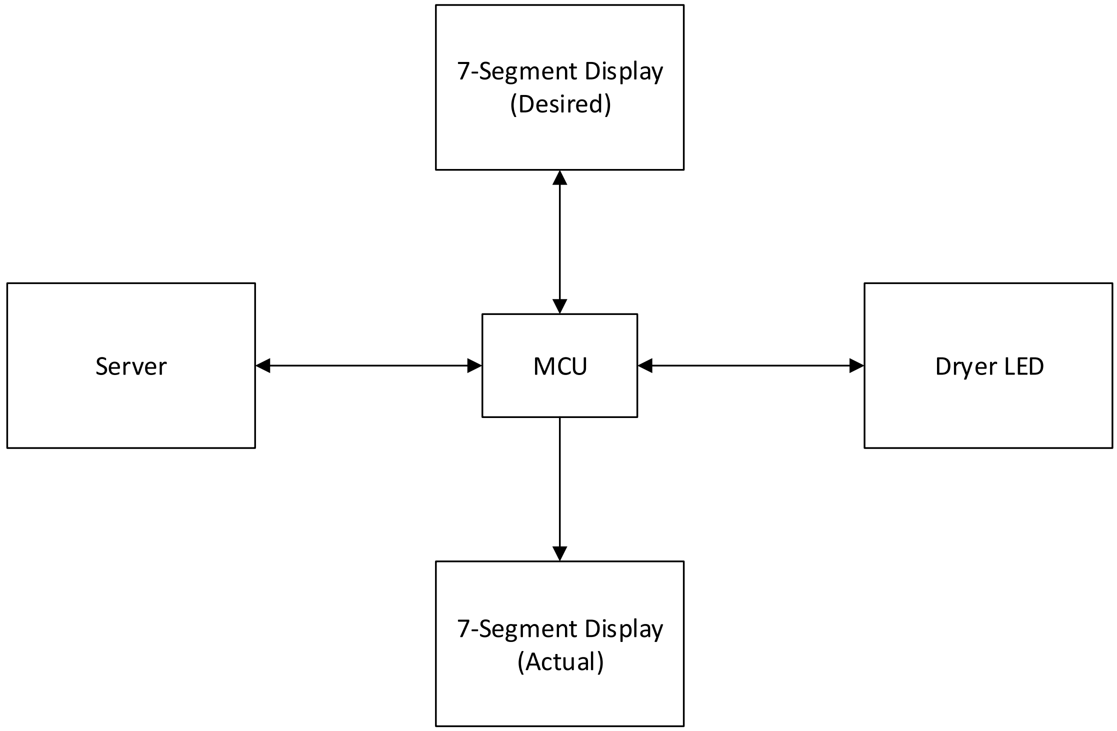


Figure 7-2: Block Diagram for HVAC/Dryer System

Our design is not intended to recognize temperature of the room or to control a home HVAC system. We will set a default "Room Temperature". How the microcontroller will recognize that the HVAC is on is by two scenarios:

- The thermostat is set to "cooling" and the desired room temperature is set lower than the actual room temperature.
- The thermostat is set to "heating" and the desired room temperature is set higher than the actual room temperature.

How the microcontroller will recognize that the HVAC is off is by four scenarios:

- The switch is set to the "off" setting.
- The desired room temperature is at the same temperature as the actual room temperature, no matter if the thermostat is set to "cooling" or "heating".
- The thermostat is set to "cooling" and the desired room temperature is set higher than the actual room temperature.
- The thermostat is set to "heating" and the desired room temperature is set lower than the actual room temperature.

How the microcontroller will command the dryer is by the following:

- The dryer will be on if the dryer switch is set to on and the HVAC is off.
- The dryer will be off in every other scenario.

The microcontroller will communicate to two different BCD-to-7 Segment Latch/Decoder/Drivers, which will communicate to the two 2-digit 7-segment displays. The BCD-to-7-Segment Latch/Decoder/Drivers we will use is the CD4511. The functional diagram for the CD4511 is shown in **Figure 7-3**.

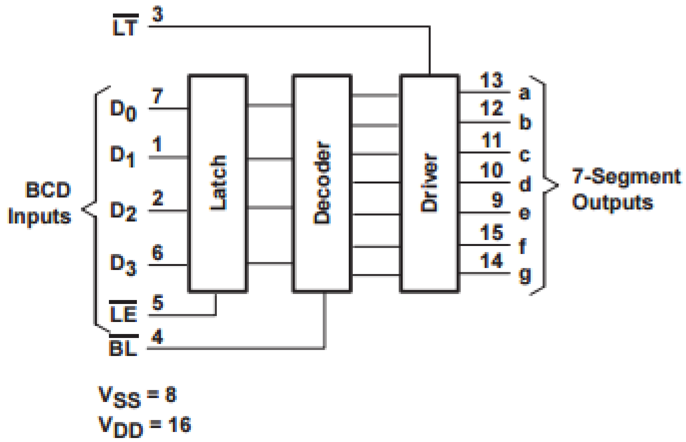


Figure 7-3: The CD4511 Functional Diagram

This chip takes a BCD input and converts it to a 7-Segment output. This output will then be connected to the 2-digit 7-segment display and will display the desired value. The necessary inputs to get certain values are shown in **Figure 7-4**.

FUNCTION TABLE

INPUTS							OUTPUTS							
$\overline{LE}$	$\overline{BL}$	$\overline{LT}$	D_3	D_2	D_1	D_0	a	b	c	d	e	f	g	DISPLAY
X	X	L	X	X	X	X	H	H	H	H	H	H	H	8
X	L	H	X	X	X	X	L	L	L	L	L	L	L	Blank
L	H	H	L	L	L	L	H	H	H	H	H	H	L	0
L	H	H	L	L	L	H	L	H	H	L	L	L	L	1
L	H	H	L	L	H	L	H	H	L	H	H	L	H	2
L	H	H	L	L	H	H	H	H	H	H	L	L	H	3
L	H	H	L	H	L	L	L	H	H	L	L	H	H	4
L	H	H	L	H	L	H	H	L	H	H	L	H	H	5
L	H	H	L	H	H	L	L	L	H	H	H	H	H	6
L	H	H	L	H	H	H	H	H	H	L	L	L	L	7
L	H	H	H	L	L	L	H	H	H	H	H	H	H	8
L	H	H	H	L	L	H	H	H	H	L	L	H	H	9
L	H	H	H	L	H	L	L	L	L	L	L	L	L	Blank
L	H	H	H	L	H	H	L	L	L	L	L	L	L	Blank
L	H	H	H	H	L	L	L	L	L	L	L	L	L	Blank
L	H	H	H	H	L	H	L	L	L	L	L	L	L	Blank
L	H	H	H	H	H	L	L	L	L	L	L	L	L	Blank
L	H	H	H	H	H	H	L	L	L	L	L	L	L	Blank
H	H	H	X	X	X	X	†	†	†	†	†	†	†	†

X = Don't care

† Depends on BCD code previously applied when $\overline{LE}$ = L

NOTE: Display is blank for all illegal input codes (BCD > HLLH).

Figure 7-4: The CD4511 Functional Table

These outputs can be achieved by using logic gates and latches, but the complexity is shown in **Figure 7-5** below. The figure represents the logic inside of the CD4511 chip.

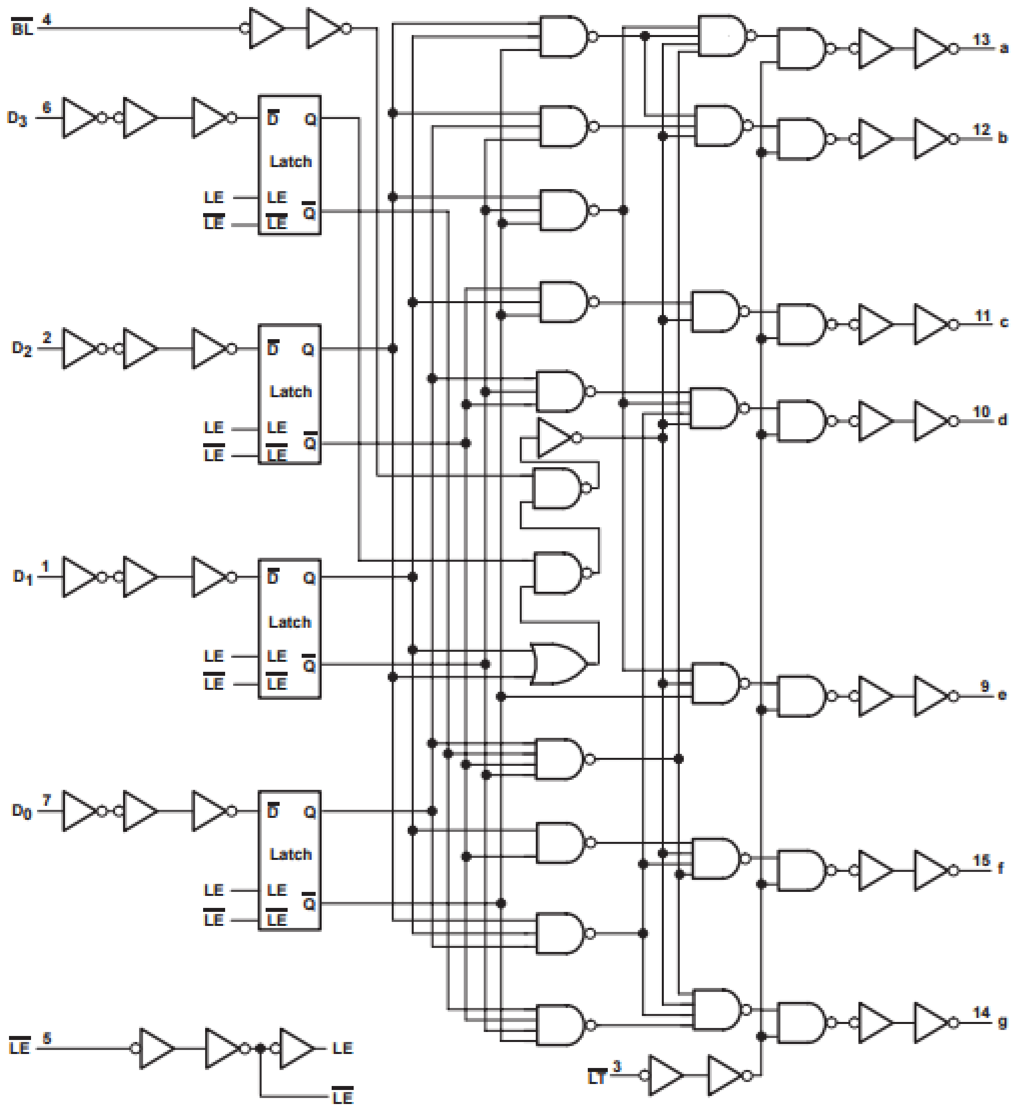


Figure 7-5: The CD4511 Logic Diagram

The CD4511 will then communicate with a 2-digit 7-segment display, more specifically, the DC58-11EWA. This display emits a high efficiency red color. The features of the DC58-11EWA are shown below:

- 0.56 inch digit height
- Low current operation
- Excellent character appearance
- Easy mounting on P.C. boards or sockets
- Two digit package simplifies alignments and assembly
- Mechanically rugged
- Standard: gray face, white segment
- RoHS compliant

The CD4511 will send signals to the various pins that represent a-g. In order to display the right number on the digit we want, we use pins 13 and 14 to activate digit two and digit one, respectively. The internal circuit diagram and pin designation is shown in **Figure 7-6** below.

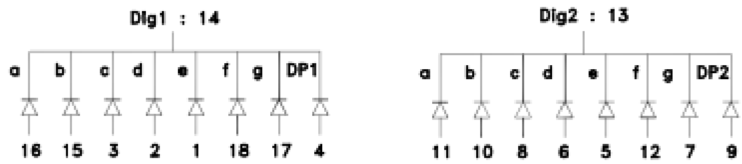


Figure 7-6: The DC56-11EWA Internal Circuit Diagram
Reprinted pending permission from Kingbright USA

We will be using the CC3200 SimpleLink microcontroller inside of our smart thermostat. This is because of its many output pins and its Wi-Fi capability. The features of the CC3200 are broken down into the three subsystems of this microcontroller (Applications Microcontroller, Wi-Fi Network Processor, and Power-Management) and are presented below:

- Applications Microcontroller Subsystem
    - ARM® Cortex®-M4 Core at 80 MHz
    - Embedded Memory
        - Ram (Up to 256 KB)
        - External Serial Flash Bootloader, and Peripheral Drivers in ROM
    - 32-Channel Direct Memory Access (µDMA)
    - Hardware Crypto Engine for Advanced Fast Security, Including:
        - AES, DES, and 3DES
        - SHA2 and MD5
        - CRC and Checksum
    - 8-Bit Parallel Camera Interface
    - 1 Multichannel Audio Serial Port (McASP) Interface with Support for Two I2S Channels
    - 1 SD/MMC Interface
    - 2 UARTs
    - 1 SPI
    - 1 I^2C
    - 4 General-Purpose Timers with 16-Bit PWM Mode
    - 1 Watchdog Timer
    - 4-Channel 12-Bit ADCs
    - Up to 27 Individually Programmable, Multiplexed GPIO Pins
- Wi-Fi Network processor Subsystem
    - Featuring Wi-Fi Internet-On-a-Chip™
    - Dedicated ARM MCU
    - WI-Fi and Internet Protocols in ROM
    - 802.11 b/g/n Radio, Baseband, MAC, Wi-Fi Driver, and Supplicant
    - TCP/IP Stack

        - Industry-Standard BSD Socket APIs
        - 8 Simultaneous TCP or UDP Sockets
        - 2 Simultaneous TLS and SSL Sockets
    - Powerful Crypto Engine for Fast, Secure Wi-Fi and Internet Connections
    - Station, AP, and Wi-Fi Direct® Modes
    - WPA2 Personal and Enterprise Security
    - SimpleLink Connection Manager for Autonomous and Fast Wi-Fi Connections
    - SmartConfig™ Technology, Ap Mode, and WPS2 for Easy and Flexible Wi-Fi Provisioning
    - TX Power
        - 18.0 dBm @ 1 DSSS
        - 14.5 dBm @ 54 OFDM
    - RX Sensitivity
        - -95.7 dBm @ 1 DSSS
        - -74.0 dBm @ 54 OFDM
    - Application Throughput
        - UDP: 16 Mbps
        - TCP: 13 Mbps
- Power-Management Subsystem
    - Integrated DC-DC Supports a Wide Range of Supply Voltage:
        - V_{BAT} Wide-Voltage Mode: 2.1 to 3.6 V
        - VIO is Always Tied with VBAT
        - Preregulated 1.85 V Mode
    - Advanced Low-Power Modes
        - Hibernate: 4µA
        - Low-Power Deep Sleep (LPDS): 250 µA
        - RX Traffic (MCU Active): 59 mA @ 54 OFDM
        - TX Traffic (MCU Active): 229 mA @ 54 OFDM, Maximum Power
        - Idle Connected (MCU in LPDS): 825 µA @ DTIM = 1
    - Clock Source
        - 40.0 MHz Crystal with Internal Oscillator
        - 32.768 kHz Crystal or External RTC Clock
    - Package and Operating Temperature
        - 0.5-mm Pitch, 64-Pin, 9-mm x 9-mm QFN
        - Ambient Temperature Range: -40ºC to 85ºC

The functional diagram for the CC3200 is shown in **Figure 7-7**.

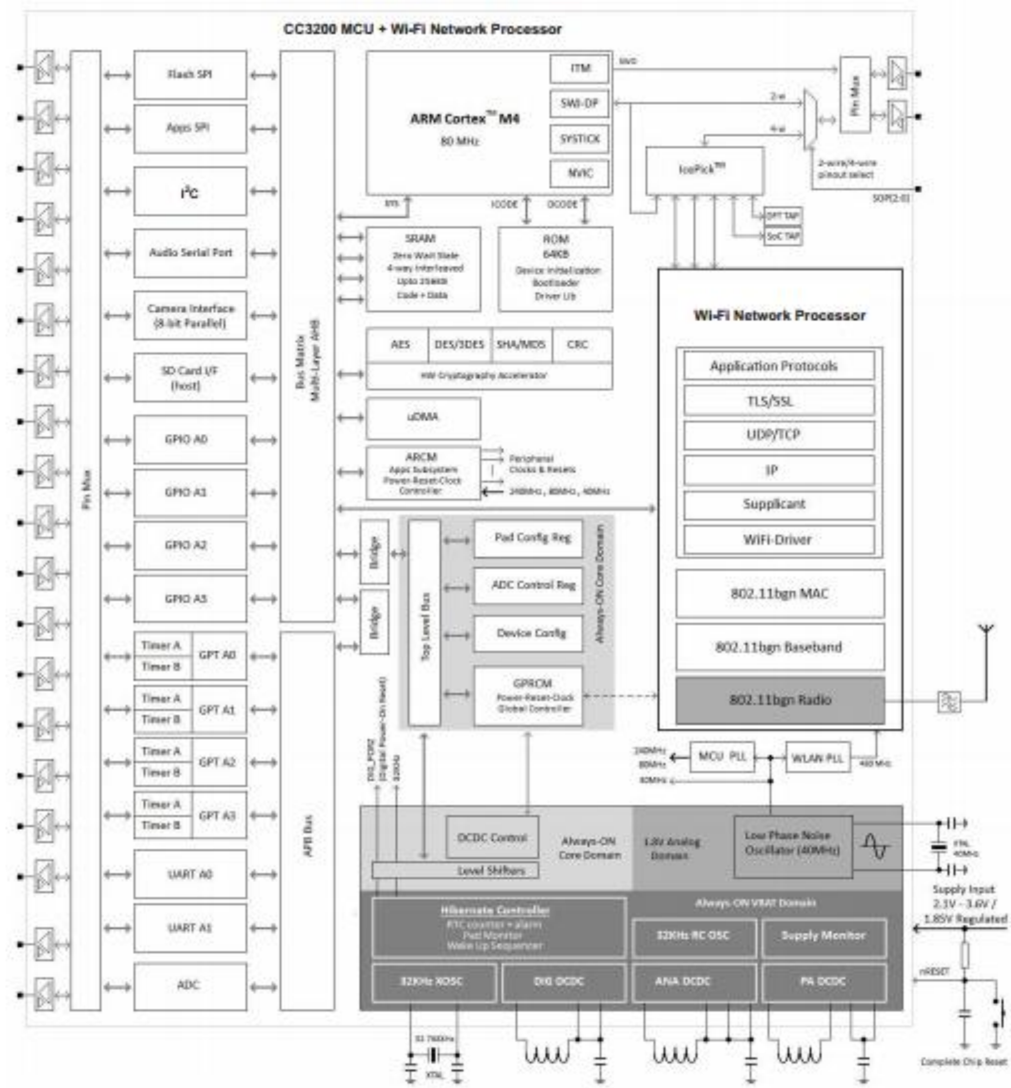


Figure 7-7: The CC3200 Functional Block Diagram
Reprinted with permission from Texas Instruments

The microcontroller will communicate with four different components, the server, the Dryer, and the two BCD-to-7 Segment Latch/Decoder/Drivers, which communicate to the two 2-digit 7-segment displays. There is a bidirectional communication between the server and the microcontroller. The server will send the user input data for desired temperature to the microcontroller, which in turn the microcontroller will change the displayed desired temperature on the display. The server will also send data to the microcontroller to either turn on or off the dryer. Finally, the server will send the default room temperature to the microcontroller. The microcontroller will send the manually changed temperature to the server, so that the user knows the set desired temperature when it is changed manually, and the microcontroller will notify the server if the dryer is switched to the on or off position. There is a unidirectional communication from the microcontroller to the actual room temperature. The actual room temperature is hardcoded. This would be different for a commercial product, obviously, because there would be sensors that would measure the temperature of the room, along with other atmospheric conditions such as humidity, but for the purpose of this project we are simply setting a standard room temperature. There is a bidirectional communication between the dryer and the microcontroller. The dryer will send a signal to the microcontroller telling whether the dryer switch is on or off. The microcontroller will send a signal to the LED representation of the dryer telling it to turn on, only if the dryer switch is on and the HVAC system is off.

7.1 LED Design

Our design also consisted of a series of a four LED array that notifies the user of how the smart thermostat is functioning. The four LEDs are used for the following:

- Normal Operation
- Sending/Receiving Data
- Problem LED
- Service LED

The Normal Operation LED will be on when the smart thermostat is functioning as designed. It is essentially telling the user that there are no problems with the smart thermostat. When the microcontroller and the server are communicating with each other, the Sending/Receiving Data LED will flash. If there is any issue with the smart thermostat there is a Problem LED that notifies the user that there is a problem, so that they can call for assistance. Finally, the Service LED notifies the user that the smart thermostat’s software is being updated. Each LED will have a different color, so that the user can identify how the device is functioning. For normal operation, the LED will be a solid green LED. For sending/receiving data, the LED will be a flashing white LED. When there is a problem with the thermostat functionality, the LED will be a solid red LED. Finally, when the thermostat is being serviced, a blue LED will light up.

7.2 Testing

Testing our product will involve testing each possible input and output for every prototype. The following table will be used during our actual prototype testing. **Table 7.2-1** below shows how we will test each prototype.

Table 7.2-1: HVAC/Dryer Testing

Prototype	Dryer Switch (ON/ OFF)	Thermostat Switch (ON/OFF)	Thermostat (Cooling/ Heating)	Desired Temp Compared to Actual Temp (Higher/ Lower)	Dryer Desired (ON/ OFF)	Dryer Actual (ON/ OFF)
1	OFF	OFF	Heating	Lower	OFF	
1	OFF	OFF	Heating	Higher	OFF	
1	OFF	OFF	Cooling	Lower	OFF	

1	OFF	OFF	Cooling	Higher	OFF	
1	OFF	ON	Heating	Lower	OFF	
1	OFF	ON	Heating	Higher	OFF	
1	OFF	ON	Cooling	Lower	OFF	
1	OFF	ON	Cooling	Higher	OFF	
1	ON	OFF	Heating	Lower	ON	
1	ON	OFF	Heating	Higher	ON	
1	ON	OFF	Cooling	Lower	ON	
1	ON	OFF	Cooling	Higher	ON	
1	ON	ON	Heating	Lower	ON	
1	ON	ON	Heating	Higher	OFF	
1	ON	ON	Cooling	Lower	OFF	
1	ON	ON	Cooling	Higher	ON	
2	OFF	OFF	Heating	Lower	OFF	
2	OFF	OFF	Heating	Higher	OFF	
2	OFF	OFF	Cooling	Lower	OFF	
2	OFF	OFF	Cooling	Higher	OFF	
2	OFF	ON	Heating	Lower	OFF	
2	OFF	ON	Heating	Higher	OFF	
2	OFF	ON	Cooling	Lower	OFF	
2	OFF	ON	Cooling	Higher	OFF	
2	ON	OFF	Heating	Lower	ON	
2	ON	OFF	Heating	Higher	ON	
2	ON	OFF	Cooling	Lower	ON	
2	ON	OFF	Cooling	Higher	ON	
2	ON	ON	Heating	Lower	ON	
2	ON	ON	Heating	Higher	OFF	
2	ON	ON	Cooling	Lower	OFF	
2	ON	ON	Cooling	Higher	ON	
3	OFF	OFF	Heating	Lower	OFF	
3	OFF	OFF	Heating	Higher	OFF	
3	OFF	OFF	Cooling	Lower	OFF	
3	OFF	OFF	Cooling	Higher	OFF	
3	OFF	ON	Heating	Lower	OFF	
3	OFF	ON	Heating	Higher	OFF	
3	OFF	ON	Cooling	Lower	OFF	
3	OFF	ON	Cooling	Higher	OFF	
3	ON	OFF	Heating	Lower	ON	
3	ON	OFF	Heating	Higher	ON	
3	ON	OFF	Cooling	Lower	ON	
3	ON	OFF	Cooling	Higher	ON	
3	ON	ON	Heating	Lower	ON	
3	ON	ON	Heating	Higher	OFF	
3	ON	ON	Cooling	Lower	OFF	
3	ON	ON	Cooling	Higher	ON	

If our testing for any prototype fails in any of these aspects we will fix said problem and retest that problem until it is fixed. Once the problem is fixed, we will retest every possible scenario again and make sure no new problem has manifested itself.

8 Web Application

As part of the overall system, the web application will mainly serve to connect the user to the system. The convenience of the web application is that it will be deployed as a website, and a mobile application. This gives the user the ability to access their account from their mobile or from a computer based on the preference and convenience of the user. Because of this, the website and app should both be consistent in the data they display as well as the settings the user configures. To make this possible, a database is necessary to consolidate the data and the settings. Both the data and the settings will be saved into the database, so that the website and app can access this data. This will ensure that the data is consistent between the website and the app.

The website and app will provide a user interface, so that the users can manipulate the settings of the system, and view the data returned from the other subsystems within the system. The other subsystems will complete the necessary calculations and tasks to realize the data that will be displayed on the website and app. The website and app will not be taking care of any of the calculations. All calculations will be taken elsewhere in the system, and will be saved into the database. The website and app will only need to communicate with the server to retrieve the data from the database, and will then display this data to the user. The website and app will also be responsible for sending the user configured settings to the server. The server will then save these settings to the database. The functionalities of the Power Strip and Above Entrance Mounted Device will depend on these user settings.

8.1 Functionalities

The website and mobile application will contain the same functionalities, which include displaying data for the users, and saving the settings specified by the users. The application will also include a process for registering the users as group members. Throughout the application, data will be stored and retrieved from the database through the server, so this section will isolate the process of storing and retrieving the data. Rather, this section will focus on the overall structure of the application, and what capabilities the user will have.

8.1.1 Data

An important aspect of the web application will be the flow of data through each of the pages and steps. The functionality of the other subsystems, including the Power Strip and the Above Entrance Mounted Device will be based on the data from the web application. The data entered by the user will be persisted in the database, and the Power Strip and Above Entrance Mounted Device will pull the data from the database, and based on the values pulled from the database, the functionalities of the Power Strip and the Above Entrance Mounted will change. Because data is such a vital part of the system as a whole, it is important to ensure that the data flow is as efficient as possible. In regards to the web application, there are to directions that the data can flow; to the web application and from the web application.

In **Figure 8.1.1-1,** the data begins at the web application, and is sent to the web server. The data can be sent to the web server in various ways, with the main way being the user changing a specific setting. The main objective of changing the settings in the web application is to change the functionality of the microcontrollers, so the main goal is for the data to reach the microcontrollers. After the data reaches the web server from the web application, the data will then be acted upon by the server, and the data will be stored in the correct tables and columns in the database. From the server, the data will be persisted in the database, to be later pulled for use in the microcontrollers. The data will then be pulled from the database and sent to the web server. At this point, the web server will have the task of sending the data to the two microcontrollers. The microcontrollers will then change their functionalities based on the data received from the microcontrollers.

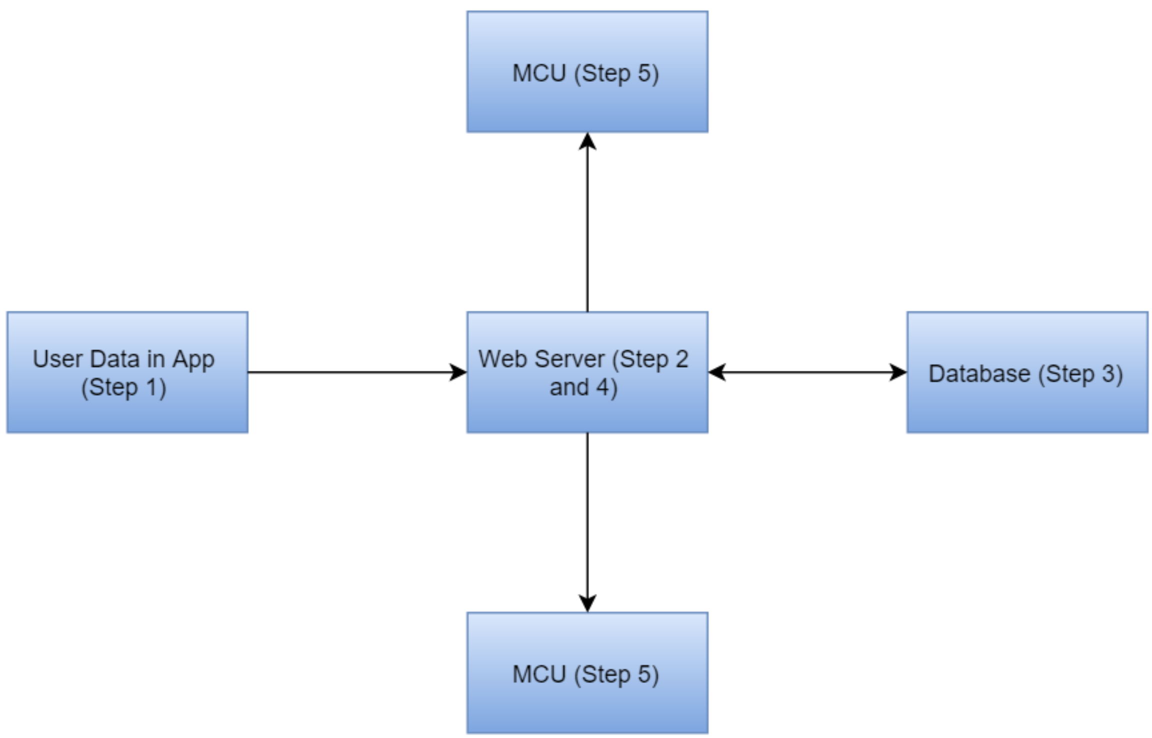


Figure 8.0-1.1-1: Data Flow Starting at Application

The data being sent from the web application can include the settings for the system, and can also include user account details. These details will typically only be persisted in the database. They will not be pulled by the web server and sent to the microcontrollers because this data will only be needed for the web application. In this scenario, the data flow will follow **Figure 8.1.1-2**, in which the data is entered by the user in the web application, and a request is sent to the web server. The web server will then validate that the data entered is the correct data that matches the data that has already been persisted in the database, in regards to the user account details. If the data input by the user passes the validations, then the web server will pull the data from the database that is associated with that specific user. That data will then be sent to the web application from the web server, and will be displayed on the user interface, for the user to either view or edit.

Figure 8.1.1-0-2: Data Entered by User

The two figure shown in this section portray the data flow to and from the web application. The web application will not be sending or receiving data directly from the microcontrollers, as the web server will be acting as a central hub for the system. The data sent to the web application will be pulled from the database through the web server and will be routed to the application. The way the web application receives data indirectly from the microcontrollers will be through the common database. The microcontrollers will send data to the web server, which will then persist the data in the database. When necessary, the web application will then request data, and the web server will send the response with the data from the database.

8.1.2 Pages

This section serves to describe the several pages that will be used in the implementation of the web application. Each sub section will explain the details of the page, including the objective of the page, and a mockup of the page in order to provide a visual understanding of the page. These sections will serve to explain the page and the data associated with the page, while the flow of the pages will be discussed in section 8.1.3.

8.1.2.1 Login Page

The first page in the web application will be the login page. This page is meant for users to login to their accounts to view the data associated with the rest of their system. The login page will also give new users the option to register a new account on the system. As shown in **Figure 8.1.2.1-1,** the user will be given the option enter a user name and password. However, if the user does not yet have an account, the user should then click on the Sign Up link, which will take the user to Register page, where the user will enter his or her information. This figure shows a mockup of the login page, to give an overall understanding of the purpose of the login page. This page can either lead to the page in which the user's details are displayed in the Building List Page, or to the Register page, in which the new user will enter his or her information.

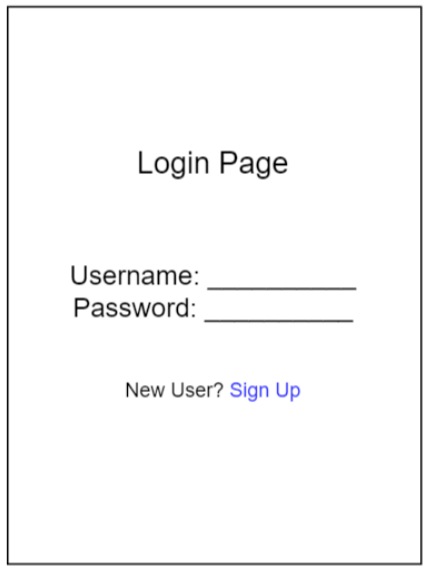


Figure 8.1.2.1-1: Login Page

8.1.2.2 Register Page

The Register Page will be the second page to appear to the user, depending on the flow that is taken by the user. In this case, it is assumed that this is a new user, and he or she does not have a registered account. This means that the new user has clicked on the Sign Up link that is available on the Login Page. This page will request the user to input a desired username, an associated password, the users email address, and the users cell phone number, as seen in **Figure 8.1.2.2-1**. The page is meant for the user to register a personal account on the system, and leads to the Group Information Page.

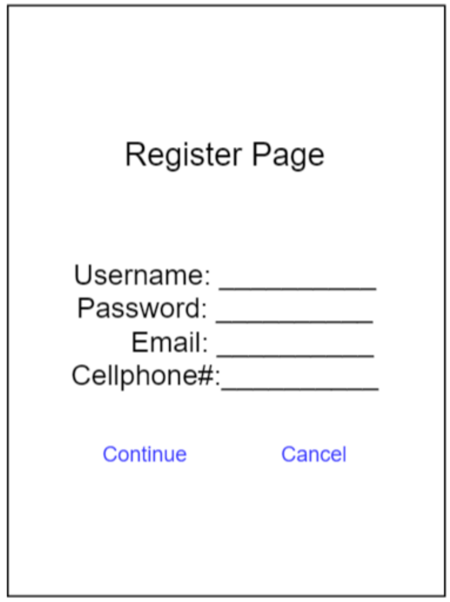


Figure 8.1.2.2-1: Register Page

8.1.2.3 Group Information Page

The objective of the Group Information Page is for the new user to join a group. The group is a collection of people that have the access and privileges to view or edit the data associated with the Power Strip and Above Entrance Mounted Device. On this page, the user should enter the group's name and the associated password, if the user would like to join an existing group. However, if the user would like to create his or her own group, the user will have the ability to do so on the Create Group Page. The mockup of the Group Information Page can be seen in **Figure 8.1.2.3-1.** The Create Group link will take the user to the Create Group Page, in which he or she can create a group.

Figure 8.1.2.3-1: Group Information Page

8.1.2.4 Create Group Page

The Create Group Page will typically be used by a user that is in the process of creating a new account and registering a new device. As part of this process, the user is initially prompted with a page, so that the user can join an existing group that uses an existing device. However, if the user does not have a group to join, the web application will take the user to the Create Group Page. This page is meant for the user to create his or her own group. The page will prompt the user to enter the group name, and the associated password. By setting this password, this will keep other users from having the ability to join any group they would like. As shown in **Figure 8.1.2.4-1,** the Create Group Page will only serve the purpose of saving the new group name and the password. This page will lead to the Building List Page, in which the buildings associated with the group are listed.

Figure 8.1.2.4-1: Create Group Page

8.1.2.5 Building List Page

The Building List Page will serve to display the buildings associated with a group. The building list will provide the ability for the user to add new buildings, and view some of the statistics associated with that specific building. The building list will also give the user to open one of the buildings from the building list, and view the list of rooms associated with the building in the Room List Page. As shown in **Figure 8.1.2.5-1,** the Building List Page will display data loaded from the database, and provide the functionality of adding a new building, and viewing and editing the settings associated with the specific building.

Figure 8.1.2.5-1: Building List Page

8.1.2.6 Room List Page

The objective of the Room List Page is to show the list of the rooms that are associated with the selected building that is associated with a specific group. The Room List Page will contain a similar concept and will have a similar to design to the Building List Page in that it will display the list of rooms within the building, and some data associated with each room. The page will also allow the user to view or edit the settings for the specific room. Similar to the Building List Page, the Room List Page provides the capability of adding a new room to be associated with the building, as shown in **Figure 8.1.2.6-1.** The Room List Page will lead to the Devices Page, which will show the devices associated with the selected room.

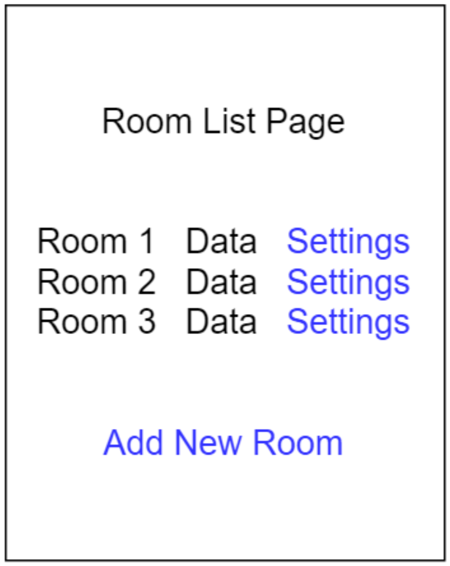


Figure 8.1.2.6-1: Room List Page

8.1.2.7 Devices Page

The Devices Page appears after the user selects one of the available rooms on the Room List Page. This page will then display devices associated with the selected room, and will display the data associated with each of the devices. As shown in **Figure 8.1.2.7-1,** the Devices Page will show the sockets on the Power Strip, and the data associated with the specific socket. This will also allow the user to view and edit the settings of the socket. The settings for the sockets would include the name of the specific socket, and the user specified settings for that socket. This can include a timer to cut off the flow to a specific socket on the Power Strip.

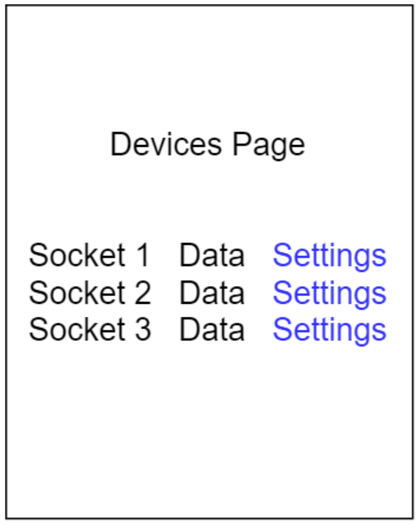


Figure 8.1.2.7-1: Devices Page

8.1.3 Page Flow

This section will serve to explain the possible paths that can be taken from the user's perspective. This will include what steps can be taken to move through the process. The flow of the data entered by the user and the data displayed to the user will be described in the next section. In certain scenarios, the flow of the data will differ based on what kind of access the users within a group are given. The flow will assume that the user has full access and privileges.

8.1.3.1 New User Flow

Figure 8.1.3.1-1 describes the page flow as well as the flow of the process for a new user that is just beginning to use the product, and creating an account for the first time on the application. Upon entry of the application the user will be prompted with a screen that asks for a username and password. However, since this flow is to describe the new user scenario, the user will click on the register button on the displayed screen. If the user already has a login username and password, the application will follow a different process. However, in this case, the application will move along the new user process and will move to the user registration page. The registration page will prompt the user to create a new account. The data requested on this page in the flow will be the username and password the user wishes to use for the account, and the user's email address. After clicking submit, this data will then be sent to the server and stored to the database. The page flow will then progress to the page, in which the user will be able to join a group.

A group is defined as a group of people that have access to view the data for specified Power Strips and Above Entrance Mounted Devices. The group could be a family, or a group of employees at a business. Administrators are defined as people within a group that have the privilege to make changes to the settings of the Power Strips and Above Entrance Mounted Devices associated with the group.

The user will be prompted to join a group by entering the group information. This flow will assume that the user does not have a group to join, and will want to create new group. The user will then be prompted to enter a group name, and a password to join the group for future users that would like to join the group.

Since the user has just created a new group, the next page of the flow will be to create a building to be associated with the group. Each group can have one or more buildings. The page will prompt the user to enter the name for the building, and the address of the building. The user will also be prompted to enter the code on the Power Strip, in order to associate the building that is being created with the Power Strip and the Above Entrance Mounted Devices. This will allow the group to have more than one buildings associated. After the building is created, the flow will progress to the building list, in which data will be pulled from the database to display the buildings associated with the group and the corresponding data. The user will then have a few options from here, including the ability to add a new building, view the default rooms associated with the building, or view the building's data. If the user selects to add a new building, the flow will be taken to a page in which data is requested from the user, in order to create the new building.

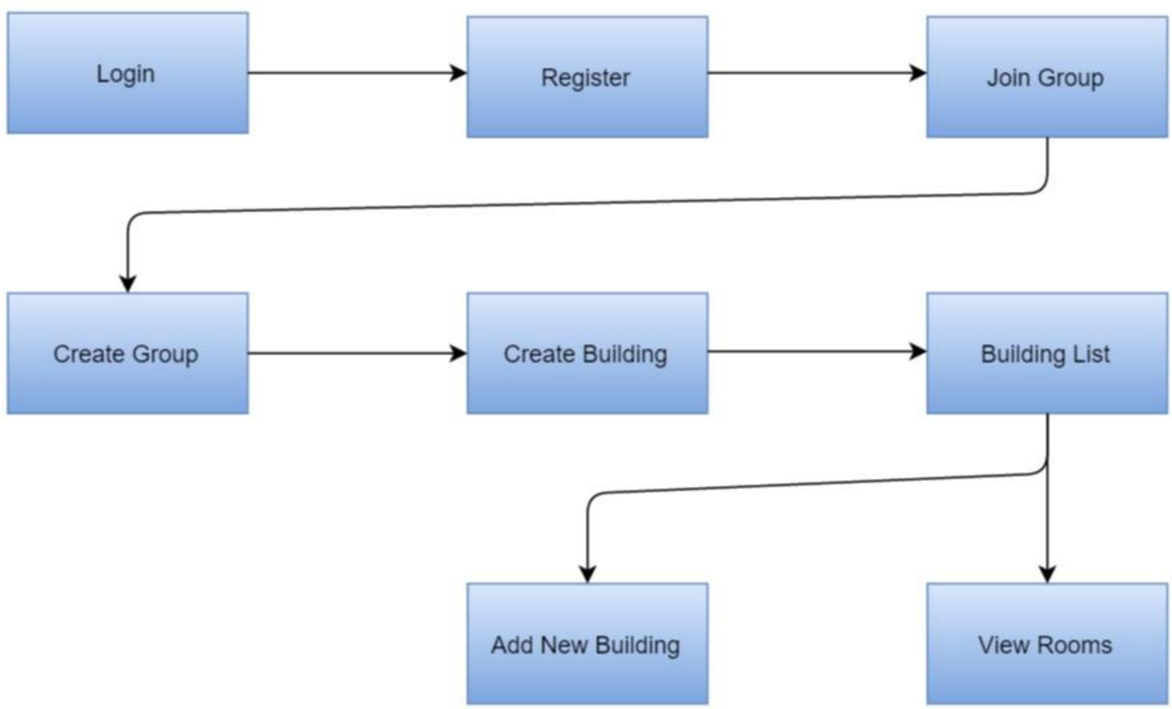


Figure 8.1.3.1-1: New User Flow

8.1.3.2 Existing Group Flow

In a similar scenario, the new user will follow the same process, however, in this scenario, the new user will be joining an existing group, which has already been through the process described above. As shown in **Figure 8.1.3.2-1,** the new user will first meet the Login Page to enter the account details. The user will then choose to register an account, and will be taken to the Register Page. On this page, the user will enter the information he or she would like to use for his or her account. The user will then be taken to the Join Group, in which he or she would enter the group name, and the password associated with the specific group, in order to be joined into the group. At this point, the data entered by the user will be validated by cross checking with the existing account data on the database. If the user has entered the correct information, he or she would be taken to the Create Building or the Building List Pages. This will vary depending on whether the buildings have already been configured for the specified group, or if they have not yet been configured. Assuming the building list has already been configured, the user will be sent to the Building List Page, and will have the opportunity to continue to the Room List Page, or to add a new building to the list.

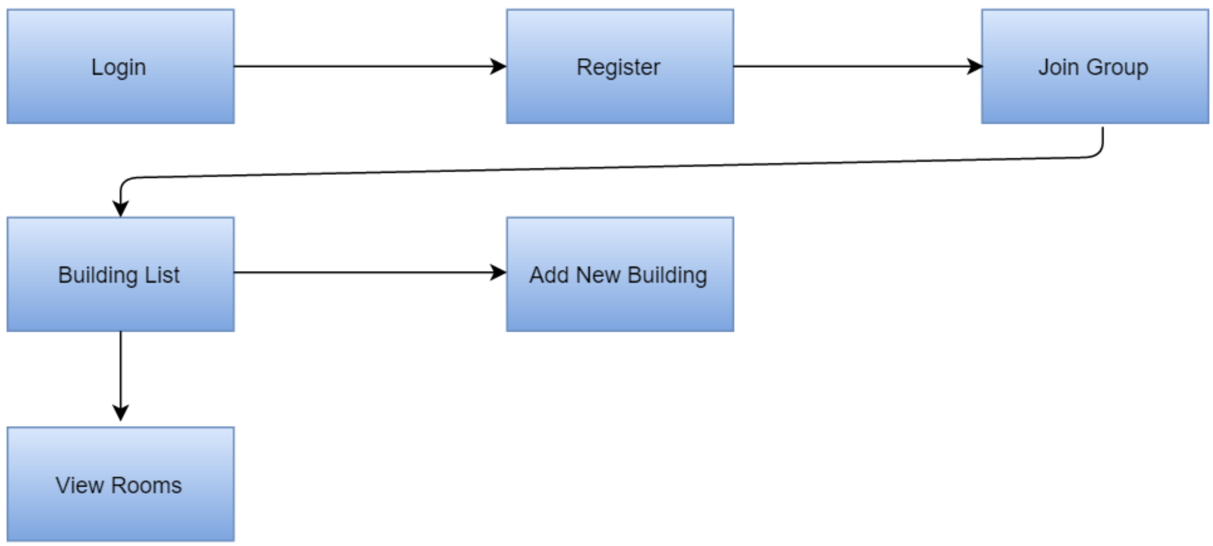


Figure 8.1.3.2-1: Existing Group Flow

8.1.3.3 Existing User Flow

In the Existing User Flow, it is assumed that the user has completely been through the New User Flow. This user has registered an account, and entered all the necessary data to register an account. The user has also either joined a group through the Join Group Page or has joined a group by creating his or her own group through the Create Group Page. The Existing User Flow assumes that the user has registered a device with his or her account, and has created buildings and rooms, if necessary. This would not be necessary if the user had joined a group that had already created the buildings and rooms. The Existing User Flow is based on the fact that the user has all the necessary data associated with the account.

As shown in **Figure 8.1.3.3-1,** the Existing User Flow will begin at the Login Page, in which the user will enter his or her username and password. The user will not need to worry about the link, in which the user will be given the ability to register a new account. The username and password entered by the user will then be validated with the data saved in the database. If the data is correct, the user will be taken directly to the Building List Page, since all the necessary data to reach this page has already been completed earlier. The user has an associated group, associated buildings, and associated rooms. At the Building List Page, the user will have the ability to follow the route in which he or she adds a new building or the route that will display the Room List Page. The route that displays the Room List Page will then allow the user to view the data and devices associated with each room by allowing the user to access the Devices Page.

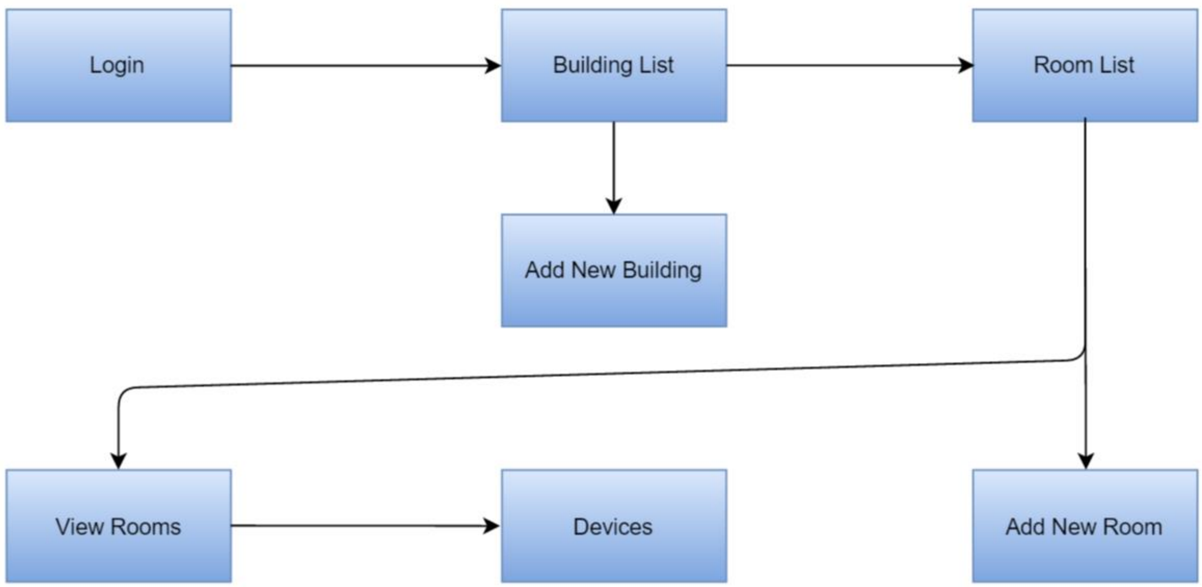


Figure 8.1.3.3-1: Existing User Flow

8.2 Development

Development of user software can be divided into two sections, front-end and back-end development, and with this distinction in classification, languages are also classified by which best accomplish these tasks. When analyzing each language, it was important to assess both the language's ability to perform what is needed and team's comfort, experience, and familiarity.

The significance of the web application will be partially reflected through the development tools and techniques used. The development tools and techniques will be such an important aspect of the system because they will lead to a more efficient system, while minimizing the development time. The development languages used will also be important because the web application will be cross platform. Therefore, it is important to note that there are certain cases where a development language works better in a certain platform than another platform. Because of this, the discussion of development tools and techniques will serve to explain how and why the decisions made were made. This section will focus on the software development of the web application, and will vary from specific programming techniques to security measures taken to improve the security of the web application. Today hardware and software are inseparable, one cannot exist without the other, and because of this a vast amount of research was needed to ensure our project's efficiency down to the lowest level of design.

8.2.1 Languages and Tools

This section will serve to explain a few of the programming languages that were examined to be used for the web application, and the decision made regarding the programming languages to be used for the web application. The languages include Java, JavaScript, XML, HTML, and CSS. This section will also serve to explain the different development tools that were examined, in order to choose a user friendly, and efficient development environment. The programming language to be chosen needed to have the ability to connect to a web server, in order to execute the server side programs that were to store and retrieve data from the database. The language also needed to support the user interface of the web application, while performing actions on the back end that is not visible to the user. This does not limit the web application to use one programming language, meaning that the web application can use a combination of programming languages to achieve this requirement. The development environment of the programming languages also needed to be simple to use, in order to avoid development complications with the environment used.

The choice of which language(s) to use can have a very large impact on a project's functionality and design, a poor decision could possibly push back a project's timeline by months or weeks, due to a teammate's inexperience or unfamiliarity, the incompatibility of the language to accomplish the assigned tasks, or an incomplete or small number of relevant libraries related to the project.

8.2.1.1 Java and XML

Our Teammates with experience in this language gained it through app development on Android Studio, and used XML as a front end language to simplify GUI design, while working in tandem with Java. XML dynamically creates user element identification numbers, is consistently updated to provide up to date GUI features, and simplifies the process of creating action and on click listeners, which drastically cuts down on code size. It was primarily created to represent arbitrary data structures, and was designed with the intention of making both a machine and human readable language to be used on internet applications. Although many of our teammates have past experience with this language, we came to a consensus that the structure of the language can make otherwise seemingly simple tasks difficult to achieve due to the unintended rigidity of the language.

All teammates had experience with this language from previous projects and coursework. Java, a class based, object-oriented language, has an overwhelmingly high number of built-in and open source libraries which can tackle a large variety of functions. Half of our team was already experienced in using Java to create GUIs for both Android and web applications. Being one of the most popular languages in use today, Java's 'write once, run anywhere' philosophy is certainly influential in the language's success. The Android SDK uses Java and XML as a basis for its applications as well as its operating system. From our past experience using Java to create GUIs, it was agreed that Java would be best used as a back-end language rather than a front-end one because of how unintuitive developing GUIs programmatically can be when simple mark-up languages can do the same job in a far simpler manner.

The first combination of programming languages examined was Java and XML. The reason the two programming languages had been combined into a package was because these were the main two used for developing an Android application. The initial aim was to develop an Android application, so Android Studio was going to be used to develop the mobile application. Android Studio is a development tool to be discussed in a later section. The use of Android Studio pushes users to program using Java and XML, since there are already many built in functionalities that can be used in both Java and XML. The main benefit of using Java as a programming language, is what makes Java a popular programming language in general; object orientation. The ability to deal with objects and attributes gives Java an advantage over other programming languages. This would have worked well with the web application, since each page within the web application can be treated as an object, or a class.

The main advantage of the Java language is the concept that Sun created, "Write once, run anywhere." The benefit of this is that the program needs to be written, and it can be run anywhere that has the Java Virtual Machine. This simplifies development, and provides the capability of reuse. The concept created by the developers of the Java language, Sun, makes Java reusable on many different systems, and throughout platforms. Java does provide the ability of developing a user interface, through which the user can interact with the web application. However, to simplify development, XML was examined to accompany Java, as the programming language to be used to develop the user interface. XML stands for Extensible Markup Language, and is mainly used to store data and allow the data to be transported. XML would be used in combination with Java to display fields on the user interface while allowing the user to either view or enter data into the fields displayed. However, when using the Android Studio, the project should use Android specific XML fields, and Java methods that are built in to improve development. Although the combination of Java and XML can be used across different platforms, using the Android Studio will make the web application an Android application only. This will limit the application from taking advantage of the full functionalities of Java and XML, and will limit the application to a single platform. Therefore, the application will not work as a website in a browser, and will only work on Android devices.

8.2.1.2 JavaScript and HTML/CSS

HTML is very widely used in web and app development because of its flexibility and ease of use, but our teams experience with the language is limited, with only one of our teammates having previous experience with it through web design. Another front end mark-up language, HTML can easily create dynamic GUI elements and unify a GUI's style and presentation with the assistance of the CSS and JavaScript languages, greatly improving the user experience. But even with its wide use and popularity, there are still issues with the language that need to be addressed. Vertical placement of elements can become a hassle due to complicated and unintuitive syntax, but oddly this is not an issue for horizontal placement. Also, there is no way to assign an element a parent or child GUI element, which would be helpful in some designs. But when viewing the HTML-CSS-JavaScript triad, the accessibility, simplicity, and separation of content and presentation make it an appealing option for app development.

The second package of programming languages examined contained JavaScript, HTML, and CSS. JavaScript is completely unrelated the Java language, and is typically found within HTML scripts. JavaScript also only runs on the browser, while Java programs are run on the Java Virtual Machine. Therefore, JavaScript does not follow the "Write once, run anywhere" concept, and depends on the browser that it is being run on. HTML is similar to this, which explains why there are certain scenarios in which JavaScript and HTML are capable of accomplishing a specific task on one browser but not on another browser. CSS is used for styling the user interface in certain ways based on the developer. Therefore, it will not be discussed as much as JavaScript and HTML. JavaScript can be written within HTML code by using the <script></script> tag in the HTML file. This provides a much simpler and cleaner technique to combine the HTML and JavaScript code. By putting the JavaScript and HTML in the same file, development and testing time will be minimized.

The combination of JavaScript and HTML will also be vital to achieve the objective of developing a cross platform application. There is no specific development tool necessary to write a program in JavaScript and HTML. All that is needed is a simple text editor to store the code, and a browser to test the program. This will provide the ability for the application to be run as a website on the user's browser. However, since this technique will not use the Android Studio for development, the web application will have to be used to generate a mobile application. Therefore, PhoneGap, a free online service will be used to take all the JavaScript and HTML code written for the web application, and generate a mobile application that can be used across platforms. Because of these capabilities and functionalities, JavaScript, HTML, and CSS were chosen over Java and XML. These programming languages will provide more functionality and flexibility, while minimizing the development time.

8.2.1.3 Development Tools

Initially, Android Studio was explored to be used as the main development tool for the web application. The use of Android Studio supports development in Java and XML. It also provides an Android device emulator to test the code written in Java and XML. Android Studio also contains the built in packages that simplify the development of an Android application. This includes the styling of the elements through the XML, and some of the built in methods that would have been used in Java. However, this issue was that the use of Android Studio would have limited the application to an Android mobile application only. Because the objective was to develop an application that can be opened in any browser, and in any mobile application, Android Studio was not chosen to be the main development tool used. It could have been used to develop the mobile application, but that would have meant that it would be required to develop an application to be displayed on the web browser.

Since the main programming languages chosen for the development of the web application were JavaScript, HTML, and CSS, the only necessary development tool was a text editor. By opening the code saved through a browser, the JavaScript program would be executed on the browser. Therefore, Notepad++ was used as the main development tool, and Google Chrome was used as the main browser for testing the programs. However, the programs will also be tested on the Internet Explorer, and Mozilla Firefox to ensure that the web application functions correctly across all browsers because there are certain functionalities supported on Google Chrome that are not supported on Internet Explorer.

8.3 Cyber-Security

Because the focus of our project was centered on energy tracking, remote control of appliances, and building-security, it meant that security in the digital realm must also be a main concern for our design to ensure user-privacy, safety, and control over their property. When assessing other designs, a shocking study by HP found that almost all Internet-of-Things security systems didn’t implement the most basic of cyber security measures. A single exploitable facet of our system, once discovered by a clever hacker puts the entirety of our system and our users at risk, so all sections from high level to low level of our design must be thoroughly scrutinized for gaps in security. Cyber security is so important because without it security and privacy cannot be guaranteed for our clients in their own personal spaces.

By examining our information systems, adding redundancy, constantly verifying proper user access, locking accounts after failed log in attempts, and alerting all group administrators whenever a new administrator has been added, implementing necessary cyber infrastructure in code to prevent or deter malicious actions and users, we believe many security oversights can be solved by the user-administrators themselves, given the proper tools.

8.3.1 User-Targeted Attacks

Specific user targeted attacks are commonly the most dangerous in the realm of cyber security, because there is a chance that it is not to collect data on an randomly chosen individual, but is actually targeted towards a specific person because of unknown motives. With the devices in place on our devices, not only would an intruder be violating our client's privacy but also the boundaries of the law. The first line in defense comes at the most basic level, password protection, and minimum password criteria.

For users, all passwords must be a minimum of ten characters long, contain, at least one uppercase letter, one lowercase letter, and one special character. With these minimum criteria, with both the length and the inclusion of all keyboard characters, passwords will be harder to discover. To guarantee private contact with our users, at registration, they will be asked to provide a private contact email, and to verify their registration by clicking the link provided in the proceeding email. This email will be our only method to contact users if their account has been breached.

8.3.1.1 Login-Access Protocol

The login page will be the gateway to our services, and without strict guidelines, would leave our users defenseless against cyber-attacks, compromising their privacy and making our devices useless. To prevent brute force methods of obtaining user's account info, accounts will be locked after three unsuccessful log in attempts, and can only be unlocked by clicking a link from an email alerting them to the log in attempt.

If multiple failed log in attempts occur within a short period of time, we will provide the user with a new auto-generated username tied to their old account in order to prevent a hacker from denying a user access to their account by continually locking it. Once logged back on, the user may create a new, personalized username in place of the auto-generated one which will hopefully safeguard their anonymity while using our service.

When failed login attempts occur, sometimes web designers with the intent of being helpful actually assist hackers in harvesting user data by specifying whether a username is not found within their system or whether a password is incorrect. To combat this issue, whenever either the username or password do not match the information held in our records, a simple statement of 'information invalid' will be displayed at the top of the screen.

If users cannot remember their login information, a request for their password and username information will be available by email if they click the 'forgot username and password' hypertext, and supply their correct contact email address. If the contact email address is correct or incorrect, the user will be treated the same regardless and an email will only be sent if the information is correct.

8.3.2 Building Group Hierarchy and Privileges

In our network of users, it was necessary to assign the role of “building-heads” to “building groups”, because giving every user access to all building information and appliance control could cause serious concern if a building group member suddenly had bad intent to disrupt regular building operations. So, the structure of user access was divided into two, regular users and administrators. Administrators are elevated group members with the ability to invite non-affiliated members into the building group, to kick building members from a building group, promote and demote building members from administrator-member positions, view power consumption statistics, and organize a building’s power schedule.

8.3.2.1 The Administrator

There must always be one group administrator, and administrators cannot demote themselves. Whenever a change in group structure occurs, whether that is promotion or demotion, or removal or acceptance of members, to ensure buildings cannot be hijacked by malicious users who have surpassed security protocol, it must be approved by all administrators. Administrators are not alerted and cannot vote when their position in the building group is being changed. It is always recommended to have at least three administrators for security purposes in the case that one of the other two administrator’s accounts have been compromised and this rogue account is trying to gain sole control of the group by demoting other administrators.

To ensure users have proper access to a page, before loading the information necessary to populate the graphical user interface, a check will occur to verify that a user is among the list of administrators, and if their username or user ID isn’t present, they will be immediately logged off and be returned to the login page. The same will be done when accessing specific buildings, if a user’s information is not found in a building group they are trying to access, presumably by exploiting a vulnerability in the system, because they have somehow surpassed previous security measures, the same protocol will be used, logging them off and sending them back to the login screen.

8.3.2.2 The General User

Non-administrator users have a limited number of abilities, because of their lower status. They have the ability to turn on and off appliances through their app, to turn on and off the building-wide security mode, and to alert users if they have accidentally triggered a security alert to be sent to other users. Beyond this, they do not have the ability to view power statistics, to alter the building power management schedule, to call authorities if an exterior entryway threshold has been broken when the building’s security mode is active, or to alter group member ship or access structure. Being a general user of a group only allows the very basic access and control of devices which can be revoked at any point by higher ranking members.

8.3.3 System Attacks

If Hackers are unable to gaining access through individual accounts their next course of action may be to bring down an entire system through attacks focused on overburdening the system infrastructure, causing the software to crash because of heavy traffic or repeated requests in a small period of time. If multiple requests are sent by a user in a very short period of time, for example three to five seconds, the user's account will be locked, their password changed, a notification email sent, and if this occurs three times within a day, the users account will be deleted for the safety of the network. If this user is the sole administrator of a building group, the group will be deleted, and all devices and information will have to be reconfigured and reentered with a new account.

8.3.4 Insecure Functions and Methods

Poorly designed, unsafe functions and methods pose very serious security risks at all levels of an application's design. By inserting unintended or very large inputs, unintended outputs can be the result, crashing the system, allowing users access to content they would otherwise be barred from, or catastrophically overwriting large sections of memory, potentially giving an unknown entity the ability to rewrite code or gain control of a device. Without knowledge of these error-prone functions cyber security and any claim of security in general becomes a farce.

In C and C++ there are many functions which handle exceptions or unexpected inputs poorly, for example strcopy(), which inputs a string and returns null if the input is too large to prevent a stack overflow. This may seem like a logical step to prevent a system from crashing, but when trying to alter or utilize that variable in another section of the function, the system will crash due to errors in handling null values. Also in C are functions which allow users to optionally specify input length, if a sack overflow occurs a hacker could covertly insert embedded code into a device or software without the user knowing, so the s* functions were created, which have the same functionality as previous functions, but have the s-prefix in front of the older functions name, denoting that the function has the ability to help developers clarify string length. The next issue with C programming is memory allocation which can also cause widespread errors in a function and memory leaks if not handled properly using the free() function. To prevent allocation issues and string length issues in C, the as* functions were created, for the same reason as the s* functions, to denote that these functions had the same functionality of older functions, but with sting length and memory allocation solutions embedded with the same functionality of the old code.

With other languages steps in defensive programming must be taken, to ensure only standard, expected inputs are given, and that unexpected inputs are in some way discarded, and not given the ability to somehow enter the system's memory, altering it and opening it up to further issues in cyber security. By being overprotective while coding, it ensures that nothing unexpected can occur. Also, robust exception handling when dealing with languages like Java is also important to implement, because unhandled exceptions, like the issues previously mentioned, can bring an entire system to a halt if an unexpected error occurs.

By utilizing defensive programming, reading and understanding the full functionality of all functions and methods used in code, handling all possible exceptions and introducing very strict and overprotective requirements of all possible inputs to the system, although a system's security can never be totally assured, team members can feel content in knowing that they made every possible attempt to protect their systems against the attacks by malicious entities to hijack or crash their system in the cyber realm.

8.4 User Experience (UX)

The user experience will be an important aspect of the system as a whole, and especially the web application because most of the interaction between the system and the user will be done through the web application. Therefore, it is important to implement user friendly features in the web application, and take into consideration the users feedback in regards to the web application. The first aspect of the user experience is the amount of data that is displayed on each page, and how much effort the user has to put in to progress through the different flows in the web application. The page flow of the application will be vital to the user experience to ensure that the user is not wasting his or her time. As part of this, there will also be links throughout the application that allow the user to jump from page to page. This will simplify the amount of work a user needs to do to switch the current page. However, by providing these links for the user, the amount of data displayed on an individual page needs to be taken into account. The page the user is currently on should not be full of data and links that will overwhelm the user, and take away from the user experience. The second aspect of the user experience will be the graphical user interface, which will simplify the amount of work the user has to complete, while also adding aesthetics to the web application. The graphical user interface will need to provide the right balance of being pleasing to the user, and providing the right amount of interaction with the user.

8.4.1 Graphical User Interface (GUI)

The Graphical User Interface (GUI) will play a large role in the user experience with the web application because the GUI will give anyone the ability to be able to use the web application. If the GUI is designed correctly, it will not matter whether the user is an expert or not, anyone would be able to learn and utilize the full functionalities of the web application. The GUI will also allow users to be more productive and efficient with their time spent using the web application. This means that the user can get more done in a smaller amount of time. To improve the design of the GUI, similar items should be placed together, and the flow of the pages should be broken down to a certain extent and ordered in the correct way. The GUI should then take advantage of the user interface elements that are provided as part of programming languages.

As part of designing the GUI, the number of pages developed that the user will flow through within the web application should not be too high. They should be broken down, but only to a certain extent. As part of finding the right amount of pages to be used for the web application, modals will be used to prompt the user to enter specific data. The modal will be discussed in more detail in a later section. Another aspect of designing an efficient GUI is using concise words to describe the fields that the user is viewing or editing. Combining this with minimizing the amount of unnecessary details on the page, will play a role in improving the GUI, as well as improving the user experience. The GUI should also display information in an ordered way, and the most efficient way to do so is by using tables, and by grouping relevant information. Therefore, the information should be sequenced in the correct way.

8.4.1.1 Elements

An element used in HTML is what is in between matching tags in the script. For example, in <p>This is an element</p>, the string “This is an element” is the element in regards to HTML. Typically, each start tag is treated as its own element. In the case of Android development, there are elements built into Android Studio that can be used through XML. The way this element is displayed to the user is dependent on which of the built in element types are used in the program. These elements may vary from a button to a dialog. In any case, the element can be treated as a property of a page that will contain a value. The value to be saved within this element can either be populated earlier, or it may be populated by the user. Elements may also contain other elements, as will be discussed in later within this section.

The first element to be discussed will be the button, as the button will be used throughout the application, and other elements in the application will use buttons. The objective of the button will be to assign a label to be displayed on the button, and to perform a specific task based on an event that occurs. In the context of the web application, the event that can occur will be a button click, which will trigger the existing code to perform a specific task. For example, if the button has the “Login” label, once the user clicks the button, the code to login to the application will be fired. Similar to the button element, link elements will also be used throughout the application. Links follow a similar functionality to buttons in that they wait for an event, the click, to fire the existing code. The reason links may be chosen over buttons or vice versa is because the use of each element will be based on what is commonly seen by users to better enhance the user experience, and user friendliness. Buttons and links can both be styled differently to change the color, border style, font design and many more to better fit the application. In the case of the web application, links will be used to flow back and forth through the pages. Therefore, the link for the page the user is currently on will be larger than the links that will take the user to the other pages. An example of a button and link can be found in **Figure 8.4.1.1-1** and **Figure 8.4.1.1-2,** respectively. The input element will also be vital in the web application as the user will be prompted to enter data throughout the whole process, especially when configuring a new account. The input element can be configured to accept normal text from the user, as well as a password, in which case, the element encrypts the password. An example of an input element can be found in **Figure 8.4.1.1-3.** Input elements can also be configured to display faded text before the user begins filling the element. This is known as a hint, and gives a hint to the user about what he or she should be inputting into the field.

Figure 8.4.1.1-1: Example of Button Element

Figure 8.4.1.1-2: Example of Link Element

Figure 8.4.1.1-3: Example of Input Element

Another element that will be part of the user interface will be the dialog element, which will be used to improve the user experience. The functionality of the dialog will be shown in the scenario in which the user clicks on a certain button, and rather than taking the user to a different page, the user will be prompted with a dialog box as shown in **Figure 8.4.1.1-4,** which shows the scenario where the user would like to add a new building to the building list that is associated with the users group**.** The dialog box will prompt the user to fill in the required information, and when the user clicks "Create," the dialog will save the information, and disappear. The significance of this is that the user will not be taken to a separate page to fill the information and then back to the initial page. Rather, the user will remain on the same page, and the dialog will be displayed as a layer on top of the current page. Once the user is done filling the information, this layer on top of the current page will be removed, and the user can continue work on the web application. The dialog will also contain input elements as well as button elements embedded within the dialog.

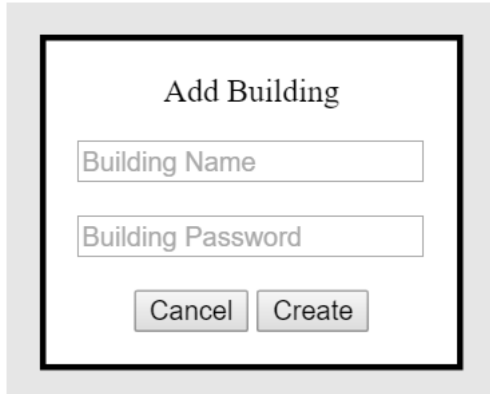


Figure 8.4.1.1-4: Example of Dialog

This section explains the main elements to be used throughout the web application, although, there will be many other elements used as part of the implementation of the web application.

8.5 Embedded Code Overview

The process of programming an embedded device must be detail-focused to ensure nothing is overlooked which could hamper its functionality. When reviewing our design we divided its tasks into two stages boot-up and normal operation which will each be described in detail below.

The boot-up and configuration process requires the device to be powered through the electrical socket which will start the boot-up process. Once boot-up is initiated the microcontroller, in station mode, will connect to the server, and continuously check the database for an account number which is associated with its device ID. The user will sync the device to their account by entering the device's ID number into a field from their app and then clicking the app's ' sync' button. The device ID is printed on the outer shell of the device. During the syncing process, if the user would prefer the device to connect to a home Wi-Fi network and disable station mode, they may do so by entering the SSID and password of the network in the fields below the device ID field before clicking the ' sync' button. If connection to this network cannot be established the MCU will send a failed connection flag which will prompt a notification to the user on their app where they will be able to repeat the syncing and configuration process.

Once the device has connected to the server and it has been synched to a user account it will then enter into a normal operating mode which will loop infinitely until it is powered off or cannot establish connection with the server. The main purpose of normal operation mode is to measure and relay power measurements from the connected appliances to the server every 1.024 seconds, and to check the 'power on' status of appliances every 4.096 seconds. The ' power on' status determines whether power should be supplied to the specified appliance through its socket. If the ' power on' status is zero, the appliance will power off and if one, power will be supplied. If connectivity is broken an interrupt will trigger and assign a value of either -1 to the intStatus flag variable which regulates the loop condition encompassing normal operation mode. Once the loop is terminated, the device will try to reconnect to its specified access point, and if unsuccessful, will return to station mode, and continue back into normal operation.

8.5.1 POWER MEASUREMENT ALGORITHM

For our design the cc3200 had to collect voltage and current samples from each socket, calculate V and I RMS values, true power, reactive power, apparent power, and the power factor. The algorithms used in the power measurement section of our device had to accommodate some of the setbacks and design of the CC3200. The analog to digital converter has a 12-bit resolution, but TI recommends to have at least a 14-bit resolution to accurately measure power. To make this possible the cc3200 sums 64 consecutive samples gathered from the ADC to get a single, more accurate measurement. With a sampling rate of 62.5 kHz for each pin, these summations, taken after sampling, will not severely distort or delay measurement samples. With a sampling time of 1.024 secs at 62.5 kHz, and having every 64 individual samples summed for greater accuracy, there will be exactly 1000 samples per pin to compare. The ADC can only measure voltages between 0V and 1.467V, to accurately translate these discretized values back into a closer to analog form, the discrete values of the ADC must first be divided by 4096 and then multiplied by 1.467. After samples are collected and processed through the ADC first V and I RMS are calculated. Using the voltage offset from the instrumentation amplifier. The Voltage coming out needs to be subtracted from the offset. The K value is the scaling factor or the gain of the op amp circuit. Then from calculating the RMS values, the algorithm calculates true power by summing the factor of voltage and current samples with the same array index value and dividing them by total number of samples taken. Apparent power is simple calculated by multiplying the I and V RMS values together. To calculate reactive power take the square root of the difference of apparent power squared and true power squared. And finally after calculating these values the phase can be calculated by taking the arccosine of true power over apparent power.

9 Database

The database will be depended on by each of the other subsystems in the system. The functionalities of each of the other subsystems will be based on data retrieved from the database, the database will have to be implemented correctly. The database will also be important because it will provide a means of consolidating all the data related to the subsystems, rather than having each of the subsystems having its own data separately. The database will also be important because it will increase the speed of the algorithms used, and improve the big O complexity. The use of a database can improve the big O complexity from $O(n^2)$ to $O(1)$. The database only needs to execute one query rather than using nested loops in the program to find the necessary value. The database will be divided into tables and columns in order to keep each value organized with a unique key for each row entered into the database.

9.1 SQL

One of the most widely used database languages, used solely to manage relational database systems, it is able to insert query, update, and delete data through declarations. Only a fraction of our team was familiar and had experience with SQL, but when viewing a teammate's code, its simplicity and readability made it clear that this was the language necessary for our project.

Although the idea of relational database being a means of establishing relations is partially correct, the relational database is mainly based on relational algebra and relational calculus. SQL is based on a stack of mathematics. It is based on relational algebra, which is built upon algebra. The relational algebra portion of the SQL is important to understand because it explains how the query should work. SQL is also based on tuple relational calculus, which is built on relational calculus. The importance of relational calculus in this case, is that it is executed through SQL declaratively rather than procedurally. This declarative processing increases the speed of the system, while also increasing the efficiency because it minimizes the amount of times needed to calculate data. This is one of the main reasons a database was chosen, as it would increase the speed and process of the system. Relational calculus is built on predicate logic, which is built on propositional logic, which is built on calculus; all of which are outside of the scope of this document. However, it is important to note that relational algebra and relational calculus are the base of SQL, and the base of relational databases, as this supports the concept of using a database for the system. The SQL used in the system will be executed through the JavaServer Pages (JSP) on the web server. These XML files will execute the correct SQL query based on the request received from either the Power Strip, the Above Entrance Mounted Device, or the app and website.

9.2 Database Configuration

The database will need to be configured on the web server, so the only way to access the database will have to be through the web server, which will be configured on the host. Once the connection is set up between the app and website, and the web server on the development end, any of the users will be able to create an account and make changes to the corresponding data in the database, as long as they have the correct privileges. The user will not need to take any actions regarding the configuration of the database, as the server and database will both be set up for the users in advance by the development team. The tables and columns in the database will also be configured by the development team beforehand, so that everything is set up for the users.

9.2.1 Privileges

An important aspect of the database is security. Data in the database will not necessarily have the ability to be accessed or altered by each of the users. For example, there will be groups who have access to the data of a specific building. However, not each of the members on the group will be able to change any of the settings regarding the building. Although all members in the group may be administrators, and have administrator privileges, if a group member is not an administrator than he or she will not have access to the data in the database. That user will not be able to update the data in the database, but will be able to view the existing data. The way to keep track of this would be to have a column of privileges in the table where user data is stored. The value of this column for the corresponding user will be checked before updating any data in database. If the privilege of the user is not at the administrator level, then an error will be returned to the user and the data will remain the same in the database. This will increase the security of the data stored in the database. The privileges will also be checked as part of the JSP pages that are executed on the server side, based on the request received from either the Power Strip, the Above Entrance Mounted Device, or the app and website.

9.3 Organization

The organization of the database becomes vital because if not done correctly, the organization of the database can cancel the benefits of using a database in the first place. The most important aspect in organizing a database is knowing the objects of the system, and knowing the properties associated with each object within the system. Each of the objects in the system will have its own corresponding table in the database, and each property within that object will have its own corresponding column in its objects table. Each table will also have a column specific for the unique key of the instance, and each row of the table will be its own instance. Organizing the database, and its tables and their columns will be essential in the implantation of the system because the functionalities of the system are based on the data stored in the database.

9.3.1 Tables and Columns

In this section, the organization and design of the database will be explained through the discussion of the design of the tables and columns within the database. This will be a similar discussion to that of the object list and each objects attributes, but will be in the context of the database. Each object within the object list will have its associated database table. Within each object and its associated database table, each of the objects' attributes will have its own associated column. The tables and columns have been designed in such a way that data can be retrieved from two separate database tables at the same time. Each row within each table will be an instance of that specific object. There are attributes that needed to be repeated throughout the objects and the tables in order to give the ability to return a single instance based on a certain scenario.

9.3.1.1 User Account Table and Columns

The first object is user account object, which will be translated to the user account table within the database, as shown in **Table 9.3.1.1-1,** which displays sample data for a few of the possible scenarios that can occur in the User Account Table and its associated columns. This will be the entity used to describe each individual user, and the attributes associated with each user. The importance of this table will be to keep track of the user information, and how the user relates to other objects and tables within the system. The attributes if the user account that will be translated to the columns within the database table will include the username and the password of the user. In this case, the username will serve as the unique identifier because only one user can have a specific username. This will be validated when the new user is creating his or her new account. The password column will mainly be used to validate the user's identity when he or she is logging into the application. The next column in this table will be the email address associated with the specific user. This will be used when sending notifications to the user based on the settings that the user has specified.

Each user will also have a specific privilege that will decide what the user can edit, and what will be read only for the user. The privilege will be saved in its own column, and contain a value of either "Admin," or "User." The admin will be given more rights than the user, such as editing settings, and modifying buildings and rooms associated with a specific group. The admin will also be able to control the users within the group. There can be more than one admin associated with a specific group. The next column will be allocated for the group username that is associated with the specific user. This will allow the development team to view a specific user, while having the ability to pull data from the Group Account table. This will associate the user with the group, and once the group data is found, the building data associated with that group can also be found.

Table 9.3.1.1-1: User Account Table Sample

Username	Password	Email	Privilege	Group Username
Joe123	Joe123	joe@joe.com	Admin	Group1
Jim123	Jim123	jim@jim.com	User	Group1
Bob	Bob	bob@bob.com	Admin	Group2
Bob1	Bob1	Bob1@bob.com	Admin	Group2
Bob2	Bob2	Bob2@bob.com	User	Group2
Bob3	Bob3	Bob3@bob.com	User	Group2

9.3.1.2 Group Account Table and Columns

The Group Account Table, as shown in **Table 9.3.1.2-1** which includes sample data for a few of the possible scenarios that can occur for this table, will be representative of the group account object. This table will be used to store the data associated with each of the groups created in the web application. The table will be used to store the username and password of each group. This can be accessed by either checking for the group username or by checking the group username that is associated with a specific user account username. This table will also contain a column for the buildings that are associated with the group. Each building associated with the group will have its own row in the table. Therefore, to access only one building associated with a specific group, the query would have to include the group username, and the building name because the group username will not be unique in the Group Account table.

Table 9.3.1.2-1: Group Account Table Sample

Group Username	Password	Buildings
Group1	Group1Pass	Building1
Group1	Group1Pass	Building2
Group2	Group2Pass	Building1

9.3.1.3 Building Table and Columns

A database table will be allocated for the building object. The Building table will be used to keep track of each instance created upon the creation of a building, as shown in **Table 9.3.1.3-1**, which includes sample data. The table will keep track of the attributes and data associated with buildings, through the use of the columns in the table. The first column in the table will be the Group Username that is associated with the specific building. This will not be unique because a group can have more than one associated building, so if the group has more than one associated building, then there will be more than one row for the group. The next column will be used to store the building name. By combining the group user name and the building name, a unique instance of the Building table can be returned. The next four columns in the table will be used to keep track of the associated location of the building. A column will be created to store the street address of the building, and another column will be used to keep track of the city, in which the building is located. Another two columns will be created to keep track of the zip code and the state that the building is located in, respectively.

Table 9.3.1.3-1: Building Table Sample

Group Name	Building	Street Address	City	Zip Code	State
Group1	Building1	123 road	City1	12345	FL
Group1	Building2	321 road	City2	54321	FL
Group2	Building1	456 road	City3	98765	FL

9.3.1.4 Room Table and Columns

The Room Table will serve to store the data associated with each instance of the rooms created by the users, as shown in **Table 9.3.1.4-1**, which includes sample data. The table will be linked to the room object which will contain data associated with each room that users create. The first attribute associated with the room will be the group name, which will be used to keep track of which group the room is associated with. This would be sufficient if we did not need to know what building the room is associated with. However, since it will be necessary to know which building the room is in, the next column in the table will be specifically for the building name. This will be followed by the column allocated for the room name. There can be more than one device associated with a room. Therefore, to get a unique room instance, the group name, building name, and room name are required. The device type associated with the specific room will be used as part of the Device table, which contains more data regarding the device type.

Table 9.3.1.4-1: Room Table Sample

Group Name	Building Name	Room Name
Group1	Building1	Room1
Group2	Building1	Room1

9.3.1.5 Device Table and Columns

The Device Table will be used to keep track of the data associated with the device object. Since each object has an associated database table, the Devices within the system will have an associated table. The first column in the Device table will be the group name in order to keep track of which group the device is associated with. For the same reason, the next column is the building name, followed by the room name that contains the device. From this table, the device within a room that is in a building that is associated with a specific group can be found. The next two columns are the product identification and the device type. The product identification will be unique to each system, and will be provided as part of the system. This code is used to synchronize the account with the device. The next column will be the device type, in which either “Power Strip,” or “Above Entrance Mounted Device” or “AEMD” will be stored. Sample data was created, and is displayed in the below **Table 9.3.1.5-1.**

Table 9.3.1.5-1: Device Table Sample

Group Name	Building Name	Room Name	Product ID	Device Type
Group1	Building1	Room1	12345	Power Strip
Group1	Building1	Room1	12345	AEMD
Group2	Building1	Room1	54321	Power Strip

9.3.1.6 Power Strip Table and Columns

The table allocated for the Power Strip object will be used to keep track of the devices within the room. The objective of the table is to keep track of which group, building, and room that the device is associated with. An example can be found in **Table 9.3.1.6-1,** which includes sample data to represent the Power Strip table. The first column will be the group name, in order to keep track of which group the Power Strip is being used by. The next two columns will be the building name and the room name to keep track of which building and which room within the building contain the Power Strip. The next columns will contain the product identification of the Power Strip which will be provided as part of the system. This is used to synchronize the group with the device to ensure that secure data transfer. The last column will be the socket identifications, which will be the sockets associated with the Power Strip, based on the settings specified by the user.

Table 9.3.1.6-1: Power Strip Table Sample

Group Name	Building Name	Room Name	Product ID	Socket ID
Group1	Building1	Room1	12345	Socket1
Group1	Building1	Room1	12345	Socket2
Group2	Building1	Room1	54321	Socket1

9.3.1.7 Socket Table and Columns

Since each Power Strip will contain sockets, the sockets will be treated as objects, and will have their own database table. A sample of the socket database table can be seen in **Table 9.3.1.7-1**. The first three columns in the Socket table will be the group name, building name, and room name. The reason these attributes are repeated between different objects and tables is because these attributes are needed to find a unique instance. These are necessary because the group name will be the only unique attribute. The next three columns will be for current measurement, voltage measurement, and power measurement, respectively. These columns will be used to store the current reading that is measured by the Power Strip.

Table 9.3.1.7-1 Socket Table Sample

Group Name	Building Name	Room Name	Socket ID	Voltage	Current	Power
Group1	Building1	Room1	Socket1	12	2	24
Group1	Building1	Room1	Socket2	6	2	12

9.3.1.8 Archives Table and Columns

The last object in the system will be the archives, so this will be associated with the Archives table in the database. The table will be meant to store older power data that is associated with a specific socket. This will contain data and be incremented as current data is moved to the Archives table. The columns in this table will allow calculations to be done based on the time period. The first four columns will be the group name, building name, room name, product identification, and socket identification, respectively. There will be twelve columns for each of the months in a year. As data becomes older, the data already stored in the Archives table will be incremented. To differentiate between years, a year column will be added, so that there will be a row for 2014 and a row for 2015, for example. These columns will only contain the power measurements, and not the current and voltage measurement. To view sample data, view **Table 9.3.1.8-1** and **Table 9.3.1.8-2**, which will be combined in the Archives table in the database.

Table 9.3.1.8-1: Archives Table Sample Part 1

Group Name	Building Name	Room Name	Product ID	Socket ID
Group1	Building1	Room1	12345	Socket1
Group1	Building1	Room1	12345	Socket1
Group2	Building1	Room1	54321	Socket1

Table 9.3.1.8-2: Archives Table Sample Part 2

Year	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
2014	300	300	300	300	300	300	300	300	300	300	300	300
2015	250	300	300	300	300	300	300	300	300	300	300	100
2015	265	300	300	300	300	300	300	300	300	300	300	80

9.4 Testing

The database testing will be have to happen throughout the life cycle of the project, starting at the beginning of the project to ensure database connectivity. To test this, the database will be opened, a test table and its columns will be created, and a test query will be run to store data into this table. Another test query will then be run on this same table to retrieve the data that was stored in the table. After this, the database will be tested through the JavaServer Pages (JSP) created on the web server. The JSP pages will contain the XML that will execute the same test queries, to ensure that the server is connected and communicating with the database correctly. The testing of the database will not be complex as the tests are only aimed to ensure that data is being stored into the database and retrieved from the database correctly. Throughout the life cycle of the project, the testing of the database will change in regards to the tables and columns used to store and retrieve data. As the project progresses the tables for other objects in the system will be created, and will need to be tested.

10 Server

To function correctly, the project will need to be able to transfer data between the main components. The microcontrollers will need to transfer data to be displayed to the user in the Android application (App) and the website, and vice versa; the app and website will need to transfer data to change the functionality of the microcontrollers because the functionality of the microcontrollers may vary based on the settings specified by the user through the app or the website. It is possible to create a direct connection between the app and website, and the microcontrollers over Wi-Fi by attaching a wireless network processor to each of the microcontrollers. However, to be able to store the data in a database, and create a network of communication between the main components, a server will be used to do so. A wireless network processor will still be required to connect each of the microcontrollers to the server over Wi-Fi.

The server will serve as the central hub of the project. It will receive data from each of the microcontrollers and store the data into the database. The server will also retrieve data from the database and send it to the microcontroller, when necessary.

It will be necessary to keep track of the settings and data obtained from the app and the website. This data will be saved into the database after being sent to the server over Wi-Fi. The importance of having this central hub is that it will consolidate all of the data, and consolidate the network links between the main components of the project. The app and website will be able to send and retrieve data to and from the microcontrollers, and the microcontrollers will be able to send and retrieve data to and from the app and the website through the use of the server and the database.

10.1 Server Types

The different types of servers need to explored, in order to find which type of server will best fulfill the goals and objectives of the project. This includes exploring the different ways to access a specific type of server, and what the server will be capable of accomplishing. There are different type of servers, including web servers and application servers. Each type of server will have different methods of access, and each type of server will have different functionalities. Depending on the type of the server, the server will also need to be hosted on a website, in order to be able to connect to the server through Wi-Fi.

10.1.1 Web Server

A web server is typically responsible for the HTTP protocol. An HTTP request is sent to the server from an external system, and the server sends the HTTP response back to the external system. The web server may also have other technologies that take care of generating the response. The response is often an HTML page to be displayed on the website or on the Android application (App). The web server chooses the best program that can generate the response based on the request received. Some of these programs include JavaServer Pages (JSP), JavaScript, and Active Server Pages (ASP). Since the main objective of using a web server would be to update or retrieve data in the database, the web server would need to choose the best of the program options to connect to the database because the web server itself cannot connect to a database. In summary, the web server is going to be responsible for receiving HTTP requests, doing some sort of processing, and sending back a response with the required data.

10.1.1.1 JavaServer Pages (JSP)

JavaServer Pages (JSP) are used when developing web applications. This can apply to the Android application (App) and the website that will be developed as part of this project. JSP pages are developed using text based documents which will deal with comprehending the request received, and generating the correct response. The JSP page is typically written in HTML and XML, however, the JSP page will also contain JSP elements, which are written in standard or XML. JSP pages have a life cycle that are controlled by the Java Servlet. The JSP page will be built automatically as part of the process. JSP pages contain a portion that is static content, which is written in HTML, WML, or XML. The static content is typically used to generate the correct response that is to be sent back to the external system. JSP pages also contain a portion that is dynamic content. The dynamic content is utilized by using properties that are associated with Java objects. As part of this, there are objects that come with the JSP technology, while other objects can be created specifically for the app and website.

10.1.1.2 Active Server Pages (ASP)

Active Server Pages (ASP) are similar to JavaServer Pages in that they are used on the server-side for web based applications. They can also be used by web servers to communicate to a database. One of the main differences is that ASP were originally developed by Microsoft, while JSP were originally developed by Oracle. ASP generates an HTML page for the web server, which is then sent to the web-based application; the Android application (App) and the website in this case. Similar to JSP pages, ASP pages use XML, HTML, or COM. The application program will refer to a .asp file on the web server, when necessary. The web server then processes the .asp file that was referred to and will execute based on what is written in the file. The ASP pages, like the JSP pages provide a means to execute commands on the database, including storing and retrieving data. The results of the ASP page can then be sent to the app and website. This will make it possible to retrieve data from the database and display it to the user.

10.1.1.3 Common Gateway Interface (CGI)

The Common Gateway Interface (CGI) adds on to the functionality of web servers by allowing the developer to specify a specific location, the CGI directory, in which the scripts are to be stored. Whenever a file within that directory is referred to, the web server will execute the script within the file. After completing the execution, the web server will then send the result of the script to the web-based application. Unlike the JSP and ASP pages which use XML and HTML, the CGI scripts use Perl to perform the tasks necessary. Another difference between CGI scripts, and JSP and ASP pages is that the CGI system which is added to the web server gives the web-based application the ability to send data to the CGI scripts by using the URL.

10.1.2 Application Server

An application server has a different objective than the web server because the application server is where the business logic for the application is kept. The business logic can then be used through the communication between the application and the application server using protocols, like HTTP. This is different than the web server that has a main objective of sending the HTML response that will be displayed to the user in the web-based application. Application servers are capable of containing a web server, so both can be used. The business logic that is kept on the application server can be accessed using Enterprise JaveBeans (EJB). Application servers provide more functionality than a database server because the database server will take care of the business logic on the client side, while the application server will take care of the business logic on the server side.

10.1.2.1 Enterprise JavaBeans (EJB)

Enterprise JavaBeans (EJB) are components kept on the application server for a specific application, and they are used to capture the business logic. An EJB will contain specifications, which are the business logic to be used by an application. Each EJB specification will contain details and directions for the application server on how the server should react and respond based on the business logic captured within the EJB. EJBs will also create the links and networks that connect the all the EJB specifications. Each EJB component is developed using a Java class and will only implement the business logic. The EJB components will be kept within an EJB container. EJB containers add functionalities that are not available through the EJB components. The EJB container will take care of how the EJB component will be saved in the database, again simplifying the tasks of the developer, so the developer will not need to worry about persisting data in the database.

10.2 Selected Server

The objective of having a server for the project will be to receive a request from the web application or the microcontrollers, process the request, and send a response back to either the web application or the microcontrollers. Because there will not be business logic, there will not be a need to keep the business logic kept on the server. Therefore, the application server will not be necessary for the project. The web server will meet the requirements of the project because it will provide the ability to retrieve and update data in the database. The web server will be kept on a host that can be accessed through the use of Wi-Fi. The web server will have JavaServer Pages (JSP) saved, and will be referenced through the web application and the microcontrollers. The reason this will fulfill the needs of the project is that the JSP pages will be programmed to process the requests sent to the server by the application and microcontrollers. The JSP pages will be composed of XML, and will either store data into the database, or it will retrieve data from the database. The data retrieved from the database will be sent to the application by the web server to be displayed to the user.

10.3 Testing

In order to test the web server, the web server should be isolated from the other subsystems in the project, except for the database. The testing of the server will include the testing of the JavaServer Pages (JSP) developed because whether the server works or not will be dependent on whether the JSP pages are working correctly. JSP pages will be creating solely for testing purposes, and will be developed to test each of the possible scenarios. To ensure that the server is working correctly, the connection has to first be established between the subsystem and the web server. Once this connection is established, then the web server passes the first test. The next test would be to ensure that the Power Strip and the web server are connected and communicating. This will ensure that the Power Strip can send data to the server and the server will store that data into the database. This will also ensure that the server can retrieve data from the database and send it back to the Power Strip. Tests will be applied to the Above Entrance

Mounted Device to ensure that the device can send data to the server and this data is correctly stored in the database. The last test will be to ensure that the app and website can send data to the server and the server will correctly store this data in the database. This test will also include ensuring that the server can retrieve data from the database and send it back to the app or website.

10.3.1 Power Strip

This section of testing procedures is meant to ensure that the server functions correctly in regards to the Power Strip. The first scenario to be tested is when data is received from the Power Strip subsystem to be stored in the database. This can include data, including the voltage and current measurements. The JSP page on the server will then execute and store the data received into the database. To ensure that this functions correctly, the power input to the Power Strip will be measured beforehand. The database will then be checked to ensure that the input power value matches the values saved into the database. If both the values match, then the server functions correctly in regards to storing the data. The next test would be to have the Power Strip request data from the server. The data would be known based on the values in the database, and can be confirmed when the Power Strip performs the correct action based on the value retrieved from the server.

10.3.2 Above Entrance Mounted Device

This section of testing procedures is meant to ensure that the server functions correctly in regards to the Above Entrance Mounted Device. The first scenario will be to ensure that the server receives data from the Above Entrance Mounted Device and correctly stores that data into the database. Since the Above Entrance Mounted Device will contain a sensor that keeps track of the room’s occupancy count, the way to test this would be to have someone walk past the device. After a person walks past it, the occupancy count of the room should change, sending a request to the server to store the new occupancy count into the database. To ensure that this functionality works correctly, the database will be checked to ensure that the occupancy count was updated and is the correct value. There will not need to be a test that checks if the device can retrieve data from the database through the server because the sole purpose of the Above Entrance Mounted Device is to update the occupancy count of the room whenever it changes.

10.3.3 App and Website

This section of testing procedures is meant to ensure that the server functions correctly in regards to the app and website. Testing of the app and website will be simpler than testing the Power Strip and the Above Entrance Mounted Device because the app and website provide a User Interface where the data will be displayed. To test that the data is sent to the server and saved correctly into the database, there will be a few use cases that simulate the user experience. The page flow will be tested, entering data and saving it through the app. To ensure that the data was saved, the same account can be logged on to through the website. If the website displays the data correctly, then that proves that the app correctly saves data, and that the website correctly retrieves data. The other aspects will be tested by doing the opposite. Data will be entered and saved through the website, and the same account would be logged on to through the app. If the app displays the data that was entered through the website, then that proves that the app correctly saves data, and that the website correctly retrieves data. These tests will show that the app and website correctly connect and communicate with the server. This will also show that the server correctly saves and retrieves data from database, and sends this data to the app and website.

11 Bill of Materials (BOM)

Table 11-1 shows our Bill of Materials (BOM). This table should represent all of the items that we will need to complete our product effectively. I have no doubt that as we progress, we will find new and/or more items that are needed.

Table 11-1: Bill of Materials

Item	Quantity	Price/Unit	Total Cost
RT314F06 Relay	3	$5.93	$17.79
ULN2003LV Relay Driver	5	FREE	FREE
LMV324 Op-Amp	38	$1.07	$40.66
TL7660 Voltage Inverter	3	$1.48	$4.40
TPS77033 Linear Regulator	5	$1.09	$5.45
Passive/Various Components	Many	Varying	$154.35
Female Socket	3	$11.37	$34.11
Extension Cord	1	$10.00	$10.00
Soldering Iron	2	$15.00	$30.00
Solder	1	$10.00	$10.00
Printed Circuit Boards	2	$66.00	$132.00
CC3200 LaunchPad	5	$30.00	$150.00
Tax	N/A	N/A	$12.52
Shipping & Handling	N/A	N/A	$301.90
		Total Cost:	$903.18

12 Project Overview

Creating our product was a big group and effort, a group consisting of two Computer Engineers and two Electrical Engineers. The process consisted of dividing the work load, creating milestones, and still working together. This process is explained in the following sections.

12.2 Project Schedule

Table 12.2-1 shows a tentative timeline of our previously achieved milestones and our future milestones. There is no assurance of these timelines, as we have never done a project of this type, size, or magnitude. We will be doing our best to meet these milestones as our time in Senior Design progresses.

Table 12.2-1: Milestones

Task	Task Length	Start and End Dates
Group Formation	2 weeks	August 24th – September 4th
Initial Project and Group Identifiaction	2 weeks	August 31st – September 14th
Boeing/Leidos Proposals	2 weeks	September 10th – September 24th
Research	3 months	September 10th – December 10th
Decide on Specific Design and Design Constraints	1 week	September 17th – September 24th
Write Research Paper	2 months	October 10th – December 10th
Submit Research Paper	1 day	December 10th
Order Parts	1 month	December 10th – January 10th
Android App	3 months	January 11th – April 11th
Current and Power Measurement Design	2 months	January 11th – March 11th
Power Strip Outlet Design	2 months	January 11th – March 11th
Relay and Relay Driver Design	2 months	January 11th – March 11th
Surge Protector Design	2 months	January 11th – March 11th
Smart Thermostat Design	2 months	January 11th – March 11th
Sensor Design	2 months	January 11th – March 11th
First Prototype	1 month	January 20th – February 20th
First Prototype Testing	1 month	February 1st – March 1st
Android Testing	2 months	February 1st – April 1st
Second Prototype	1 month	February 20th – March 20th
Second Prototype Testing	0.5 months	March 1st – March 20th
Final Design	1 month	March 1st –

		April 1st
Final Design Testing	0.5 months	March 20th – April 1st

12.3 Task Management

We broke up the work load in a way that would pertain to our differing majors. The two Computer Engineering (CE) majors (Gabriel Holland and Omar Mohammed) worked mainly on the software side of our design, and the two Electrical Engineering (EE) majors (Tyler Ensey and Spencer Sullivan) worked mainly on the hardware of our design. This does not mean however that the software team won't be doing work on the hardware side, and that the hardware team won't be doing work on the software side. Doing work for the separate teams will help in expanding our knowledge and give us a chance to get a look in on the all-around product.

12.3.1 Division of Labor

We broke up the work below for the hardware and software teams. Each Item, whether it be hardware or software, has a primary engineer and a secondary engineer. The division of labor is shown in **Table 12.3.1**.

Table 12.3.1-1: Division of Labor

Component	Primary Engineer	Secondary Engineer
Socket Design	Spencer Sullivan	Tyler Ensey
Relay and Relay Driver Design	Tyler Ensey	Spencer Sullivan
Power and Current Measurement Design	Spencer Sullivan	Tyler Ensey
Surge Protection Design	Spencer Sullivan	Tyler Ensey
Power Strip Printed Circuit Board	Spencer Sullivan / Tyler Ensey	N/A
Power Strip Microcontroller Programming	Gabriel Holland	Omar Mohammed
Web App Programming	Omar Mohammed	Gabriel Holland
Database Programming	Omar Mohammed	Gabriel Holland
Server Programming	Omar Mohammed	Gabriel Holland

13 Project Conclusion

We look forward to materializing our product. First, it will consist of the main component, a smart power strip. This power strip measures power consumption of each outlet on the power strip and sends this information to the server and, thus, the user. Also, this power strip will allow for the user to cut off power to any of the three outlets when desired. This gives full functionality of power management to the user, thus cutting the user's power bill. Another feature was the above entrance mounted device, this recognizes activity within the home. If there is nobody inside a room with a smart power strip, the power strip will cut power to all of the outlets.

This feature can also be bypassed by the user if they do not want power cut to certain outlets. Finally there was the feature that was beneficial to the power companies, the HVAC and dryer system. This system did not allow for the HVAC system and dryer to be on at the same time. Overall, we are very pleased with our project decision, it is a great project for increasing our knowledge in both the Electrical and Computer Engineering fields.

14 Usage

The functionality of our design goes through the server. Every change, reading, and user specification has to proceed through the server. To use the smart power strip, the user must first plug it into a power source, that source being a wall outlet for a residence or building with functional power. The user then has full control over the device. The user starts with the web app, creating a “profile” by entering their desired “Username”, “Password”, “Email”, and “Cellphone#”. The user can then create a group name and password for the group for the power strip. This group can then be joined by anyone that has authorization to control the power strip. There can be multiple power strips in multiple different locations that the user would like to control. The user can keep all their power strips into one group. There is then a page for buildings. The user can have multiple buildings that they have power strips housed. The user can add new buildings at any time. The user can select the “Settings” link, which allows the user to change the name of the building. For example, they can name the building “Home”, “Work”, “Vacation Home”, etc. Then, for each building there is a room list. The user can select the “Settings” link, which allows the user to change the name of the room. For example, they can name the room “Bedroom”, “Living Room”, “Kitchen”, etc. Also, the user can add a new room at any time. Then finally, there is a devices page. This page has each socket within the room. The user can name each socket however they would like, or leave it as the default. The default is “Socket 1”, “Socket 2”, and “Socket 3” for each power strip. If the user would like to rename the socket, they can name it whatever they would like. For example, “Television”, “Phone Charger”, “Desktop Computer”, etc. For each socket there is a “Settings” link, this link allows for the user to set the settings for each socket. These settings include the ability for the user to tell the socket to turn off if the room occupancy is zero or to tell the socket to be on/off at all times. Each socket will also have a data section that shows the kWh consumption for each socket. The user will have the option of adding a $/kWh specification. From this inserted data, the monthly power bill will bill be calculated for each socket. Thus, the monthly power bill for each power strip will be calculated. Thus, the monthly power bill for each room will be calculated. Then, finally, the monthly power bill for each building will be calculated. Figure 14-1 shows the web app that a user will utilize to control and read data.

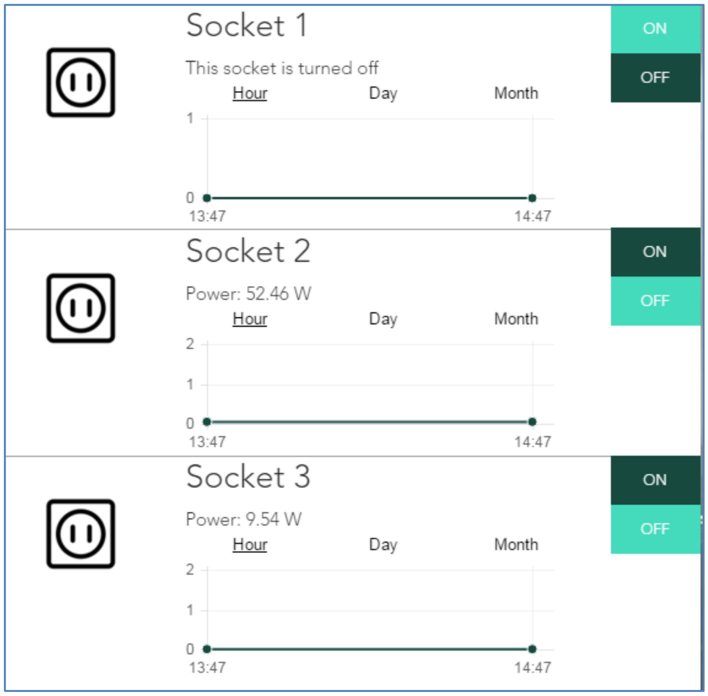


Figure 14-1: Energy Guard Web App

Each socket can be turned on or off by toggling the on or off button on the web app. The power consumption shown is what is currently being used by the smart power strip. The graph is what will display energy usage by hour day or month. The smart power strip must be plugged into a wall outlet using 120 V AC Voltage at 60 Hz.

15 Project Final Testing

The final product was tested extensively by using different common household appliances plugged into the power strip. We used a fan, Lamp, and a soldering iron. Each outlet had to provide the power needed for the appliance to work and the consumption posted to the server and read back on the web app within a tolerance of +/- 5 W. The Power strip itself was tested by making sure each circuit gave us the offset and voltage needed so the MCU could read the signal properly. This was accomplished by using a DMM on the test points we put on our PCB. Current was read as a voltage by the MCU and converted by the MCU using the equations in this document. The product worked as intended after we tested each socket with a DMM and Looking at what the web app gave us. The testing was final when our read values matched the values reported by the web app within a tolerance of 5 W.

Appendices

Appendix A – Copyright Permissions

Permission to use Figure 4.2-1

nest

Hi ,

Thanks for contacting us with your Nest question.

We'll reply within 1-2 business days. But if you need help right away, please call us at 855-4MY-NEST (855-469-6378). Refer to your email address or case number 01689290 when you call so we can quickly find your information.
We are open 24/7.
Thanks for being a Nest customer,
The Nest Support Team

Your original question:
Hi, my name is Tyler Ensey. A few other students and I are doing a senior design project that involves creating a smart thermostat, and we used the Nest Learning Thermostat as inspiration for our design. I was wondering if it would be alright for us to use an image of the Nest Learning Thermostat, on your site, in our project report? Thank you.

Permission to use Figure 5.6.1-1

Copyright Permissions (3) People

Tyler Ensey <tylerensey@yahoo.com> Today at 1:44 PM
To smawebmaster@sharpsec.com

To Whom it may Concern,

My name is Tyler Ensey, and I am in my senior year at UCF. My senior design project includes using one of your devices in the design of a smart power strip, I was looking for permission to include your functional diagram of the S108T02 SSR in my research paper. If you can respond ASAP, I would greatly appreciate it.

Sincerely,

Tyler Ensey

Reply Reply to All Forward More

Lavallee, Don <LavalleeD@sharpsec.com> Today at 1:50 PM
To Tyler Ensey
CC SMA Webmaster

Mr. Ensey, the product you reference in email to our web master is discontinued. The product documentation (graphics and product details) available on web site, may be used in your research paper. http://www.sharpsma.com/download/s108t02-epdf

Regards,

Don LaVallee
Director - Strategic Business Operations
Sharp Microelectronics of the Americas
1.360.834.8089 office I 1.360.773.5372 mobile I 1.360.834.8653 fax
lavalleed@sharpsec.com

Please consider the environment before printing this e-mail.

Permission to use Figures 5.6.1-2 and 5.6.1-3 (Pending)

Tyler Ensey <tylerensey@yahoo.com> Today at 1:51 PM
To info@ixys.net

To Whom it may Concern,

My name is Tyler Ensey, and I am in my senior year at UCF. My senior design project includes using one of your devices in the design of a smart power strip, I was looking for permission to include your functional diagrams of the CPC1998 and CPC1965Y SSRs in my research paper. If you can respond ASAP, I would greatly appreciate it.

Sincerely,

Tyler Ensey

Reply Reply to All Forward More

Permission to use Figure 5.7-1 (Pending)

Permissions to use Pictures

SS Spencer Sullivan
To: info@iec.ch; Reply all | Wed 12/9/2015 1:05 PM

To whom it may concern,

My name is Spencer Sullivan. I am a student at the University of Central Florida. I am currently working on my Senior Design project for Electrical Engineering. I would like to ask permission to use two pictures from your IEC website in my senior design project report. The two pictures I am referring to are of IEC Type A and Type B sockets. The pictures are located at the following webpages. The content will be credited to IEC and its website in the paper.

Type A: http://www.iec.ch/worldplugs/typeA.htm
Type B: http://www.iec.ch/worldplugs/typeB.htm

Thank you,
Spencer Sullivan
spencer.sullivan@knights.ucf.edu

Permission to use Figure 7-6 (Pending)

Tyler Ensey <tylerensey@yahoo.com> Today at 5:51 PM
To sales@KingbrightUSA.com

To Whom it may Concern,

My name is Tyler Ensey, and I am in my senior year at UCF. My senior design project includes using one of your devices in the design of a smart power strip, I was looking for permission to include your internal circuit diagram of the DC56-11EWA 7-segment display in my research paper. If you can respond ASAP, I would greatly appreciate it.

Sincerely,

Tyler Ensey

Reply Reply to All Forward More

Permission to use Figures from Texas Instruments Datasheets

Appendix B – References

[1] Adafruit Industries, 2014, "PIR Motion Sensor", from http://learn.adafruit.com/downloads/pdf/pir-passive-infrared-proximity-motion-sensor.pdf

[2] Adobe Systems, 2015, "Step 1: Install PhoneGap", from http://docs.phonegap.com/getting-started/1-install-phonegap/desktop/

[3] Advanced Circuits, 2015, "Designing for Easy PCB Assembly", from http://www.4pcb.com/pcb-assembly-design.html

[4] Advanced Circuits, 2015, "Engineering Student Program", from http://www.4pcb.com/pcb-student-discount.html

[5] Bulgin, "IEC Connectors", from http://www.mouser.com/ds/2/57/In-Outlets-2010-13960.pdf

[6] Coding Geek, 2015, "How Does a Relational Database Work", from http://coding-geek.com/how-databases-work/

[7] Constine, Josh, 2014, "Who Gets Rich From Google Buying Nest? Kleiner Returns 20X On $20M, Shasta Nets ~$200M", from http://techcrunch.com/2014/01/13/nest-investors-strike-it-rich/

[8] CUI Inc., 2015, "AC-DC Power Supply", from http://www.cui.com/product/resource/vsk-s2.pdf

[9] Daly, M., Associated Press, 2012, "Poll: Americans Know How to Save Energy, But Bulk", from http://www.apnorc.org/news-

media/Pages/News+Media/poll-americans-know-how-to-save-energy-but-balk.aspx
[10] Digital Zoo, 2014, "The Importance of Great User Interface Design", from http://www.digitalzoo.co.uk/blog/item/the-importance-of-great-user-interface-design
[11] Express PCB, 2015, "Manufacturing Services", from http://www.expresspcb.com/miniboard-standard/
[12] Fitzpatrick, Jason, 2012, "The How-To Geek Guide to Measuring Your Energy Use", from http://www.howtogeek.com/107854/the-how-to-geek-guide-to-measuring-your-energy-use/
[13] Gurari, Eitan, "Relational Calculus", from http://web.cse.ohio-state.edu/~gurari/course/cse670/html/cse670Ch15.html
[14] Hamby, Chris, 2014, "'Clean energy' Expansion Could Strain US Power Grid", from http://www.publicintegrity.org/2011/02/04/2155/clean-energy-expansion-could-strain-us-power-grid
[15] Harris Interactive, 2015, ""No Matter the Method, Fewer Americans Taking Steps to Improve Energy Efficiency at Home", from http://www.harrisinteractive.com/NewsRoom/HarrisPolls/tabid/447/mid/1508/articleId/1569/ctl/ReadCustom%20Default/Default.aspx
[16] IEC, 2015, "Appliance couplers for household and similar general purposes - Part 1: General requirements", 60320-1:2015
[17] IEC, 2015, "World Plugs and Sockets", from http://www.iec.ch/worldplugs/typeB.htm
[18] IEEE, 2012, "Wireless LANs", 802.11
[19] iFixit, 2015, "Nest Learning Thermostat 2nd Generation Teardown", from https://www.ifixit.com/Teardown/Nest+Learning+Thermostat+2nd+Generation+Teardown/13818
[20] IXYS Integrated Circuits Division, 2015, "CPC1965Y AC Solid State Relay", from http://www.clare.com/home/pdfs.nsf/0/F8D486C47F61006E85256A2C0069F23A/$file/CPC1965Y.pdf
[21] IXYS Integrated Circuits Division, 2015, "CPC1998 AC Power Switch", from http://www.clare.com/home/pdfs.nsf/www/CPC1998.pdf/$file/CPC1998.pdf
[22] Jacobe, D., Gallup, 2013, "Americans Want More Emphasis On Solar, Wind, Natural Gas," http://www.gallup.com/poll/161519/americans-emphasis-solar-wind-natural-gas.aspx
[23] Jansen, B. J., 1998, "The Graphical User Interface: An Introduction", SIGCHI Bulletin, **30**(2), pp. 22-26.
[24] Java, "How is JavaScript different from Java?", from https://www.java.com/en/download/faq/java_javascript.xml
[25] Java, 2010, "The Java EE 5 Tutorial", from https://docs.oracle.com/javaee/5/tutorial/doc/bnagx.html
[26] Kezunovic, M., Dobson, I. and Dong, Y., 2008, "Impact of Extreme Weather on Power System Blackouts and Forced Outages: New Challenges", from http://iandobson.ece.iastate.edu/PAPERS/kezunovicBPC08.pdf

[27] Kingbright, “14.22mm (0.56INCH) DUAL DIGIT NUMERIC DISPLAY”, from http://www.mouser.com/ds/2/216/DC56-11EWA-43385.pdf
[28] Lashinsky, Adam, 2014, “Is Tony Fadell the next Steve Jobs or ... the next Larry Page?”, from http://fortune.com/2014/06/12/tony-fadell-nest/
[29] Lehner, Martin, 2014, “The Importance of a Good GUI”, from http://www.geekbraindump.com/2014/06/03/the-importance-of-a-good-gui/
[30] Microsoft Corporation, 2000, “Active Server Pages Tutorial”, from https://msdn.microsoft.com/en-us/library/ms972337.aspx
[31] Nest Labs, 2015, "Nest Learning Thermostat", from https://store.nest.com/product/thermostat/
[32] Nest Labs, 2015, “How do I update the software on my Nest Learning Thermostat?”, from https://nest.com/support/article/How-do-I-update-the-software-on-my-Nest-Learning-Thermostat
[33] Nest Labs, 2015, “How does Nest Protect connect wirelessly?”, from https://nest.com/support/article/How-does-Nest-Protect-connect-wirelessly
[34] O’Reilly & Associates, 2001, “Key Benefits of Java”, from http://docstore.mik.ua/orelly/java-ent/jnut/ch01_02.htm
[35] OKSolar, 2015, “Technichal LED’s Color Chart”, from http://www.oksolar.com/led/led_color_chart.htm
[36] Reuters News, 2014, “Kleiner Perkins, Shasta Ventures make about 20x their money on Nest Labs – Reuters”, from https://www.pehub.com/2014/01/kleiner-perkins-shasta-ventures-make-about-20x-their-money-on-nest-labs-reuters/
[37] Rouse, Margaret, 2008, “TCP/IP (Transmission Control Protocol/Internet Protocol) definition”, from http://searchnetworking.techtarget.com/definition/TCP-IP
[38] Sensirion, 2014, “Datasheet SHT20”, from http://www.mouser.com/ds/2/682/Sensirion_Humidity_SHT20_Datasheet_V3-469724.pdf
[39] Sharp, 2004, “S108T02 Series/S208T02 Series”, from http://cdn.sparkfun.com/datasheets/Components/General%20IC/S108,208T02%20Series.pdf
[40] Sintes, Tony, 2002, “App server, Web server: What's the difference?”, from http://www.javaworld.com/article/2077354/learn-java/app-server-web-server-what-s-the-difference.html
[41] STMicroelectronics, 2015, “Ultra-low-power 32-bit MCU ARM®-based Cortex®-M3, 128KB Flash, 16KB SRAM, 4KB EEPROM, LCD, USB, ADC, DAC”, from http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00277537.pdf
[42] Suehring, Steve and Valade, Janet, 2015, “How to Organize the Data for Your MySQL Database”, from http://www.dummies.com/how-to/content/how-to-organize-the-data-for-your-mysql-database.html
[43] Texas Instruments, 1998-2003, “CD54HC4511, CD74HC4511, CD74HCT4511 BCD-TO-7 SEGMENT LATCH/DECODER/DRIVERS”, from http://www.ti.com/lit/ds/symlink/cd74hct4511.pdf

[44] Texas Instruments, 1999-2015, “LMP848x Precision 76-V High-Side Current Sense Amplifiers With Voltage Output”, from http://www.ti.com/lit/ds/symlink/lmp8481.pdf
[45] Texas Instruments, 2006-2008, “TPL9202 8-CHANNEL RELAY DRIVER WITH INTEGRATED 5-V LDO AND BROWN-OUT DETECTION”, from http://www.ti.com/lit/ds/symlink/tpl9202.pdf
[46] Texas Instruments, 2009-2015, “INA28x High-Accuracy, Wide Common-Mode Range, Bidirectional Current Shunt Monitors, Zero-Drift Series”, from http://www.ti.com/lit/ds/symlink/ina283.pdf
[47] Texas Instruments, 2011-2015, “INA226 High-Side or Low-Side Measurement, Bi-Directional Current and Power Monitor with I 2C Compatible Interface”, from http://www.ti.com/lit/ds/symlink/ina226.pdf
[48] Texas Instruments, 2012-2015, “DRV120 Power-Saving Current Controlled Solenoid Driver”, from http://www.ti.com/lit/ds/symlink/drv120.pdf
[49] Texas Instruments, 2012-2015, “ULN2003LV 7-Channel Relay and Inductive Load Sink Driver”, from http://www.ti.com/lit/ds/symlink/uln2003lv.pdf
[50] Texas Instruments, 2013-2015, “CC3200 SimpleLink™ Wi-Fi® and Internet-of-Things Solution, a Single-Chip Wireless MCU”, from http://www.ti.com/lit/ds/symlink/cc3200.pdf
[51] Texas Instruments, 2014, “CC3200 SimpleLink Wi-Fi and Internet-ofThings Solution, a Single Chip Wireless MCU Technical Reference Manual”, from http://www.ti.com/lit/ug/swru367b/swru367b.pdf
[52] Texas Instruments, 2014, “Three-Outlet Smart Power Strip”, from http://www.ti.com/lit/ug/tidu453/tidu453.pdf
[53] Texas Instruments, 2014-2015, “CC3100/CC3200 SimpleLink™ Wi-Fi® Internet-on-a-Chip User's Guide”, from http://www.ti.com/lit/ug/swru368a/swru368a.pdf
[54] Texas Instruments, 2014-2015, “CC3200 SimpleLink™ Wi-Fi® and IoT Solution, a Single Chip Wireless MCU Programmer's Guide”, from http://www.ti.com/lit/ug/swru369b/swru369b.pdf
[55] Texas Instruments, 2015, “MSP432P401R LaunchPad™ Development Kit”, from http://www.ti.com/lit/ug/slau597a/slau597a.pdf
[56] Texas Instruments, 2015, “MSP432P401x Mixed-Signal Microcontrollers”, from http://www.ti.com/lit/ds/symlink/msp432p401r.pdf
[57] Texas Instruments, 2015, “MSP432P4xx Family Technical Reference Manual”, from http://www.ti.com/lit/ug/slau356a/slau356a.pdf
[58] Texas Instruments, 2015, “SN74LV4051A 8-Channel Analog Multiplexers and Demultiplexers”, from http://www.ti.com/lit/ds/symlink/sn74lv4051a.pdf
[59] Transfer Multisort Elektronik, 2015, “Bulgin PC0675/PC”, from http://www.tme.eu/en/details/px0675_pc/ac-powerconnectors/bulgin/
[60] United States Department of Labor Occupational Safety & Health Administration, 1981-2007, “Electric Utilization Systems”, 1910.302
[61] US Energy Information Administration, 2012, “Residential Energy Consumption Survey Data Show Decreased Energy Consumption Per Household”, from http://www.eia.gov/todayinenergy/detail.cfm?id=6570

[62] US Energy Information Administration, 2013, “Two Perspectives on Household Electricity Use,” from http://www.eia.gov/todayinenergy/detail.cfm?id=10251
[63] UT News, 2014, “UT Energy Poll and OurEnergyPolicy.org Surveys Reveal Support For Energy Efficiency and Concerns About Political Squabbling”, from http://news.utexas.edu/2014/04/30/ut-energy-poll
[64] W3Schools, 1999-2015, “Introduction to XML”, from http://www.w3schools.com/xml/xml_whatis.asp
[65] Winkler, Rolfe and Wakabayashi, Daisuke, 2014, "Google Pays $3.2 Billion for Nest Labs," The Wall Street Journal, p. B1.