School of Engineering Science
Simon Fraser University
Burnaby, BC V5A 1S6
cmeerker@sfu.ca

November 12, 2015
Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University
Burnaby, BC. V5A 1S6

Re: ENSC 440W Design Specification for “SimpleHome”

Dear Dr. Rawicz,

Please find the attached design specification for @HOME’s SimpleHome project. SimpleHome is a home automation solution with the goal of bringing intelligence and simplicity to the home automation market at a low cost. The design specification details the implementation of this project's components, the SimpleHome Hub, peripheral devices and interfaces, detailing the rational and chosen implementation for each part of the system.

This specification specifically focuses on the proof-of concept stage of the project. It also details the processes that will be used to test and validate the system. If you would like more information or have any questions regarding the design specification, you can contact me via e-mail at cmeerker@sfu.ca.

Sincerely,

Curtis Meerkerk

@HOME : SimpleHome

Design Specification

Project Team:
Curtis Meerkerk
Daniel Quon
Ekta Sachdev
Kara Imhof
Meghan Lui
Nas Makkiya

Primary Contact:
Curtis Meerkerk
cmeerker@sfu.ca

Submitted to:
Andrew Rawicz - ENSC 440W
Steve Whitmore - ENSC 305W
School of Engineering Science
Simon Fraser University

Issue Date:
November 12, 2015
Revision:
1.6

Abstract

The idea of the smart home has been around for quite some time. One example being Jim Sutherland's ECO IV made in 1966 which computerized household chores [1]. Now you can buy "smart" devices in most hardware stores and the market is expected to explode [2]. The problem is that the technology like in 1966 is more of a novelty than an asset. The typical system requires substantial time to setup and create rules to operate. There is also the question of whether it actually saves labour. One case of this is that it takes longer and more effort for the user to get their phone and go through a menu in an app to turn on the lights in a room than walking to the switch on the wall. For home automation to succeed it must become a tool that is beneficial to the user not another novelty item.

Our goal is to make a home automation system that doesn't need rules or hours of setup. We aim to provide a way to make home automation simple. Throughout the following design specification there are four core values that set the design apart: Simple, Efficient, Inexpensive and Secure. The goal is to take home fixtures like a light switch, power outlet or blinds and add a wireless switching mechanism that receives commands from a central hub. The hub is designed to function on startup independent of the user collecting data and making decisions based on that data. The user is also able to interact with the hub via an online interface though it is more intended to be informative and to remotely check-up on the house.

The final key component to the design is the market demand which for another novelty item is rather low. However, with the increasing concentration of seniors there is a demand for system that would allow family or caregivers to check-up on independent seniors. This is where SimpleHome fits perfectly it is a simple system that sets itself up and connects with familiar fixtures. It allows a secure connection to check on the occupant and best of all it is designed to be inexpensive. We on the @HOME team believe that this design shows SimpleHome to be the system to meet the needs of this rapidly expanding market and a competitive one at that.

Table of Contents

List of Figures

Glossary

Intel Edison	A small computer system that runs a specialized Linux Distro communicating via Wi-Fi and Bluetooth [3].
MongoDB	An open source database that is document oriented [4].
Venetian Blinds	Common household blinds with 4 definitive states (up, down, open closed) Up/Down is controlled by a cord which stack slats together. The Open/Close motion is controlled by a rotating rod which varies the tilt of the horizontal blinds changing the amount of light passing through. [5]
Wi-Fi	A Wireless Fidelity composed of local area wireless computer networking technology allowing electronics devices to network [6].
ZigBee	A specification for high level communication between devices wirelessly [7].

List of Acronyms

ABS	Acrylonitrile butadiene styrene
API	Application-programming interface
CSA	Canadian Standards Association
GUI	A Graphical User Interface
HPPE	High-Density Polyethylene
HTML	Hypertext Markup Language
HVAC	Heating, Ventilation and Air Conditioning
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
PLA	Polylacitide
RF	Radio Frequency
RSS	Radio Standards Specification
SSR	Solid State Relay
SoC	System-on-Chip
THWN	Thermoplastic Heat and Water-resistant Nylon-coated
W3C	World Wide Web Consortium

1. Introduction

1.1. Background

The SimpleHome endeavour was undertaken with the goal of providing an inexpensive, energy-efficient, secure and easy-to-use solution for the growing home-automation industry. As a team of Systems and Electronics Engineers, our skill-set was put to optimum use during the product development phase of this product. @HOME’s aim is to design a system, which would augment the daily life of the user, by reducing the stress that comes with maintaining a smart home.

A major demand for a reliable home-automation system can be found amongst independently living seniors. It is expected that a quarter of the population of Canada will be seniors, in about 15 years [8], adding immense stress on the old-age home-care system. In such a scenario, our product comes with the assurance that a user’s home can be controlled by using a simple and accessible website and app.

This design specification document outlines how each component of the SimpleHome system comes together to make a practical home-automation system which is simple-to-use, energy-efficient, uses inexpensive components and places maximum priority on user-data security.

1.2. Scope

The designs specified in this document pertain to the proof-of-concept stage for the project and the validation testing of the system. This document explains how the functional specifications are met for the minimum viable product, along with some details for the progression into the alpha stage of the project's functionality.

1.3. Intended Audience

The functional specification document is intended to be used by all individuals who are part of the dynamic team at @HOME. This document is anticipated to guide the team through the design and development stages of the product, as well as serve as a reference when formulating the progress report for the project. Additionally, it is expected of each individual to refer to this document during test-plan generation and user manual documentation phases. The test-plans outlined in this document—will be executed during the testing phase—in order to crosscheck the operability of individual components, as well as, the combined home automation system.

2. System Requirements

2.1. System Overview

An overview diagram of the design goals for the SimpleHome product is shown in Figure 1. @HOME intends to create a smart home automation system which can be remotely controlled by users. The controlling gears of this system lie in the SimpleHome Hub, which serves as a medium between the user commands and the actual devices being operated in the household. A detailed design analysis of SimpleHome Hub's hardware and software components is provided in the subsequent sections.

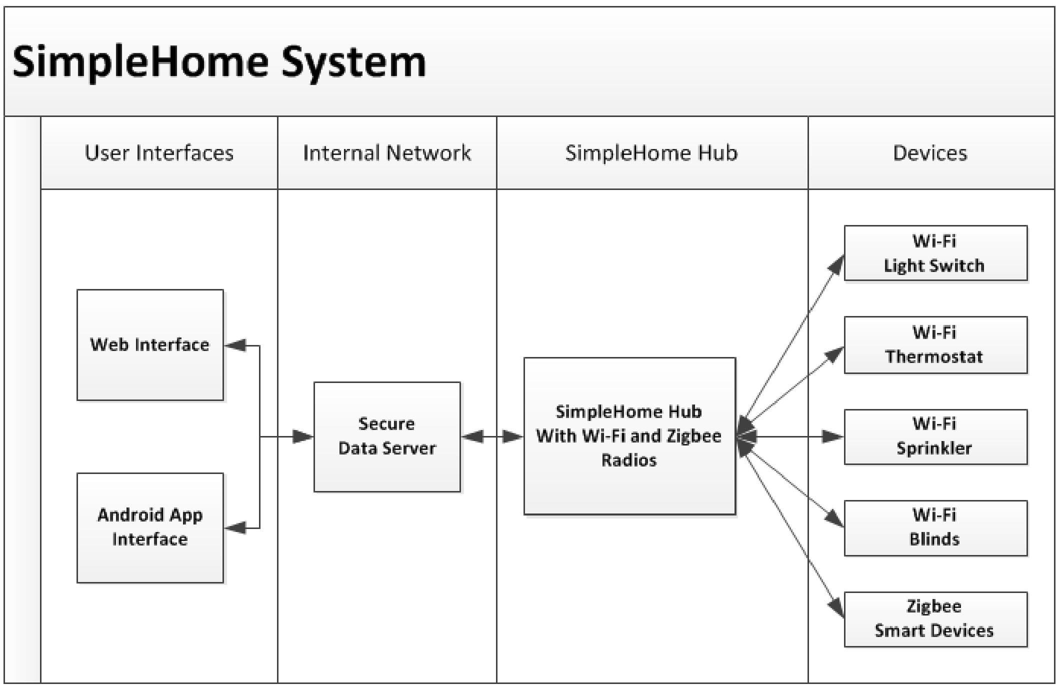

Figure 1: High-Level System Overview

2.2. Communication System Design

The SimpleHome Hub is equipped with wireless communication tools which allow it to serve as the medium between the remote user and the peripheral devices placed in the house. As shown in Figure 2 below, the Hub uses Wi-Fi and ZigBee communication protocols to build its own network of devices to interact with. Additionally, the Wi-Fi capabilities of the Hub, allows user data to be backed up on the user-accessible data server.

Figure 2: User-Hub-Devices Communication System Overview

2.2.1. BeagleBone Black

The BeagleBone Black is a low-cost single-board computer, which serves as the control-center in the SimpleHome Hub design. The key functionalities offered by this board are shown in Figure 3.

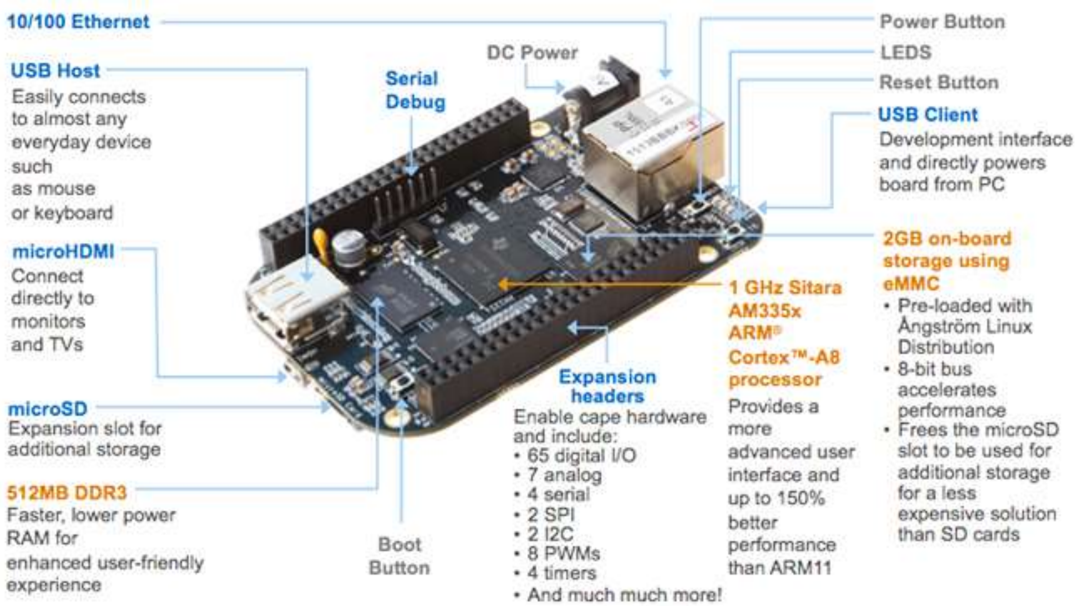

Figure 3: Capabilities Overview of the BeagleBone Black [9]

There were a number of reasons why the BeagleBone Black was chosen as the core element of the Hub. Since the board can be purchased for $55 USD, it provides us with a wide range of functionalities for a relatively low cost. The board comes built in with a full Linux Debian distribution, which will serve as the base OS for our product, due to its ease-of-use, high compatibility with wide-range of peripherals, and large community support base.

The additional ports on the BeagleBone Black are exploited to serve various needs of our complex system. The USB port serves as the medium for building the connection to the ZigBee-enabled devices in the household. The presence of the Ethernet port in addition to Wi-Fi radio provides an easy alternative solution to connect to home router. Additionally, the BeagleBone Black is easy to add functionalities to by using custom capes. One such custom cape is the Wi-Fi cape used by the SimpleHome Hub to interact with the Wi-Fi-enabled devices in the user's home.

2.2.2. ZigBee Connectivity

The SimpleHome Hub is equipped with a wireless transceiver system to transmit and receive data using the ZigBee communication protocol. The CC2531EMK ZigBee USB Module, as shown in Figure 4, was selected to serve this purpose.

Figure 4: CC2531EMK ZigBee USB Module [10]

The CC2531EMK is a USB-enabled SoC solution for ZigBee communication needs. It is a certified IEEE 802.15.4 radio transceiver module [10]. The relatively small size, ease-of-use and large-scale applicability of this module is the reason why it is chosen by most small-scale developers working in the ZigBee interfacing field.

2.2.3. Wi-Fi Connectivity

Most of the peripheral devices will be controlled by the SimpleHome Hub using the Wi-Fi communication protocol. In order to establish the connection between the Hub and the peripheral devices, a Wi-Fi cape will be added onto the BeagleBone Black. Figure 5 shows the WL1835MOD w/ chip antenna Cape, incorporated into the design of the SimpleHome Hub. The WL1835MOD chip on this Wi-Fi Cape supports the IEEE 802.11bgn standard [11]. This cape also has Bluetooth capabilities built in, and in further production revisions of our product, we can exploit this capability to support additional custom devices.

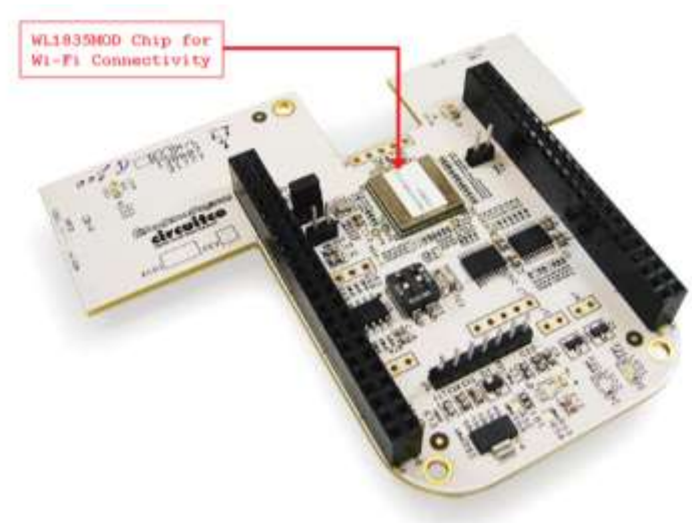

Figure 5: BeagleBone Black Wi-Fi Cape with WL1835MOD w/ chip antenna [12]

We chose to use a Wi-Fi cape instead of the more easily available USB Wi-Fi Dongles as the only USB port is used by the CC2531 ZigBee module. Another alternative would be to purchase a USB extender and incorporate the two wireless communication protocols via USB. However, this would cause interference between the ZigBee and Wi-Fi connectivity and would land up being a more cumbersome alternative.

2.3. Enclosure Design

The SimpleHome Hub is the driving force behind our whole home automation system design as such it must be reliable for the system to succeed. Therefore, it is necessary that the Hub casing is immune to dust interference, temperature variations and physical damage. Additionally, since our target audiences include the elderly and the physically-disabled, the enclosure design is smooth and free of sharp edges. The Hub is also designed to be wall mountable or can sit on a flat surface. Figure 6 below highlights our safe and robust proof-of-concept enclosure design for the SimpleHome Hub.

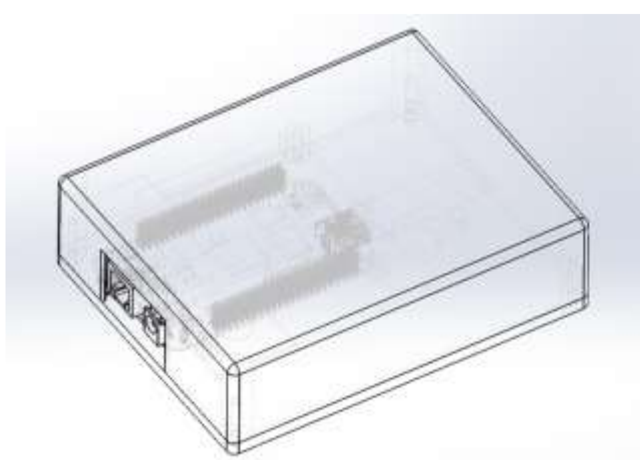

Figure 6: Proof-of-Concept SimpleHome Hub Enclosure Design [13]

Since the SimpleHome Hub is aimed to be a stationary system, it was necessary for the enclosure to be sleek and unobtrusive in nature. The above shown design is also very cheap and environment-friendly, as we aim to 3D-print this enclosure using inexpensive, recyclable plastic. Using recyclable plastic also ensures that the recyclability cradle-to-cradle design aspect is met.

3. Peripheral Devices

3.1. Overview

In designing the SimpleHome system it was challenging to find peripherals to use with the system as retail smart home systems typically work on a proprietary protocol or a closed source API. So in the spirit of modularity we divided the task of connecting with the existing products and designing our own. The choice was made early on to connect with existing Zigbee products being the more open of the proprietary protocols and having a large market share [14]. The design of our own peripheral devices is to show the potential of this project to control the lighting and temperature of the house in a simple and inexpensive way. This area is the most extensible in this project with the potential of adding integration with other protocols and existing devices in the IoT space. The following designs are our building blocks for different peripherals. The peripherals are designed to be modular, meaning that the various sensors cam be combined with any switch based peripheral to create a multi-function peripherals.

3.2. Light Switch

Turning on and off light switch is one of the simplest functions of a home automation system and therefore one of earliest ways to verify functionality the system. As such, the ability to turn on and off a lightbulb is the primary test of functionality for both Zigbee and Wi-Fi to validate the proof-of-concept design. Since there are many existing Zigbee light options and with the goal being to show that the SimpleHome hub connects to existing Zigbee products the choice was made to use the GE Link bulb to test the Zigbee connection to the Hub. Additionally, we have designed a Wi-Fi connected light switch which uses the ESP8266 SoC containing a Wi-Fi radio to control a triac diode used to switch power going to the light. The design is to replace an existing electrical light switch and add one with a Wi-Fi connection along with the desired sensors for light or temperature.

3.3. Electrical Power Outlet

Automating the use of our appliances like switching a light is a fundamental feature of any well respected home automation system, as they are used to either control the amount of idle power consumed, or to control the power state of various devices. For the SimpleHome system to control power in a simple way, we designed a wall plug (See Figure 7). The wall plug is connected to an existing power outlet and then like the light switch uses the ESP8266 SoC to control a switch, in this case a Solid State Relay (SSR), to toggle the power to whichever device is plugged into the wall plug. The SSR is used to safely switch a 10A current load of an attached device such as an air conditioner or Space Heater.

The schematic diagram for the switching circuit used in both the @HOME light switch and electrical power outlet is detailed in Appendix 0 .

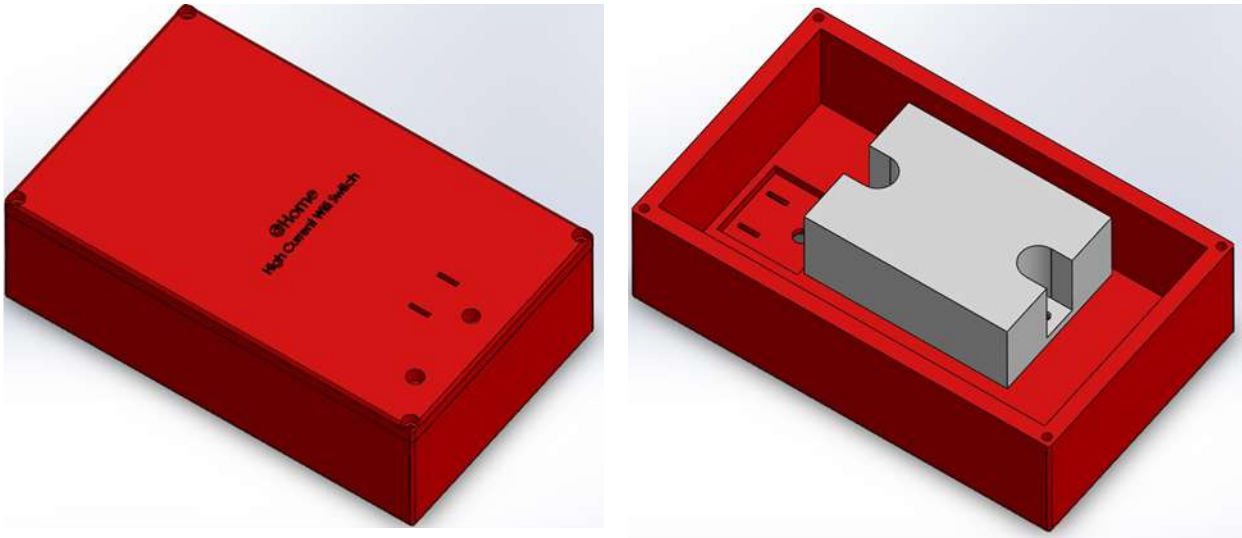

Figure 7: Wall Plug Enclosure

3.4. Temperature Sensor

Sensing the temperature is one of the key requirements of any home automation system. By using a simple one wire temp sensor connected to a SoC we can easily monitor temp in any part of the house. The temperature peripheral again uses the ESP8266 SoC for Wi-Fi and to communicate with the one wire temp sensor.

The schematic diagram for the temperature sensor is detailed in Appendix 0, Figure 19.

3.5. Thermostat

As there are several existing options that wirelessly and intelligently control HVAC systems as such, we took one of the more popular models the Honeywell FocusPro Wi-Fi Thermostat and designed our system to work with this specific model for the proof-of-concept stage.

3.6. Light Sensor

Sensing the amount of light in a room is really useful telemetry to control the amount of light in the room whether to turn on more light or to open the blinds. The temperature peripheral uses the ESP8266 to read the voltage value of the light sensor.

The schematic diagram for the light sensor is detailed in Appendix 0, Figure 18.

3.7. Automated Blinds

For the purpose of demonstration a set of Venetian blinds will be procured and a wooden frame will be built. The entire apparatus will fit within an 80cm x 60cm x 50cm (height x width x depth) area. Figure 8 outlines the mechanical specifications for the automated blinds.

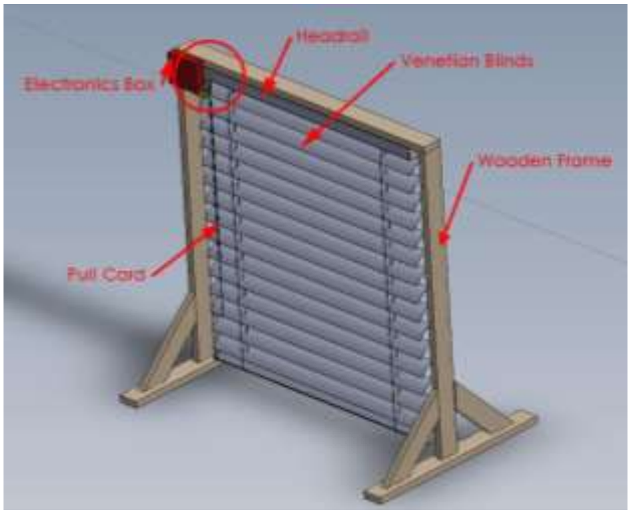

Figure 8: Physical Design of Automatic Window Blinds

By removing the tilt rail included in the assembly of the blinds we are left with a small eyelet attached to the head rail which when rotated controls the tilt of the blinds. Our 5V motor, when attached to the eyelet can be used to control the tilt of the blinds. Figure 9 is a close-up of the mechanical parts of the system.

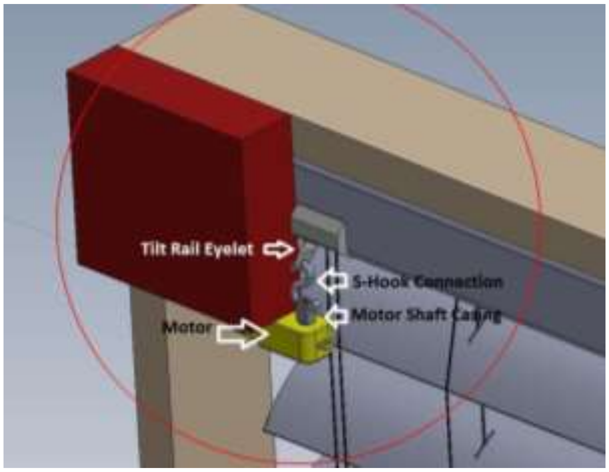

Figure 9: Mechanical Components of Automated Blinds

Inside the component box will be an Arduino microcontroller that will take inputs from the photocell, temperature sensor and HUB to produce an output on the motor. Power will be provided to all components via a DC Jack which will be connected to the AC Adapter and wall plug. The Photocell and Digital Temperature Sensor included in the system assist the SimpleHome Hub to decide what state the blinds are in and if they need to be opened or closed. To send and receive data from the Hub a Wi-Fi module will be attached to the Arduino.

3.8. Firmware

The firmware of all the previous peripherals will follow a simple command and response API, the firmware. The firmware opens a TCP port number 43333, and listens for specific commands, for binary switching it listens for SWITCHx=y, y is for the binary switch state, and x is the switch number in that peripheral. For temp/light peripheral the peripheral listens for “GETTEMP” for temp and “GETLIGHT” for light. Then the peripheral responds with the sensor value.

3.9. System Learning

In order to make the system smart the goal is to add machine learning algorithms to run on the data. This is to allow the SimpleHome Hub to run without the need for the user to setup and live by a set of programmed rules but rather to have the hub determine the rules based on the user. This is achieved by using support vector machines to classify the data into various areas and using neural network methods to find patterns of behaviour in the data. The proof of concept system is designed to categorize the data and to detect patterns. The machine learning algorithms were trained on the “2010 ZigBee Sensor Data Collected from a House (in Vancouver, Canada) Dataset” collected by Stephen Makonin [15]

4. User Interface

4.1. Overview

The user interface will be built and designed using Meteor, a JavaScript framework integrated with MongoDB. The database will act as our centralized platform to store and synchronize peripheral information. As such, the user interface and BeagleBone Black will be subscribed to the database in order to allow the user to control the BeagleBone Black.

4.2. Front-End: Website Design

The website for the UI will be divided into 4 separate web pages for each peripheral type from the homepage including Lighting, Temperature, Security and Other as seen below.

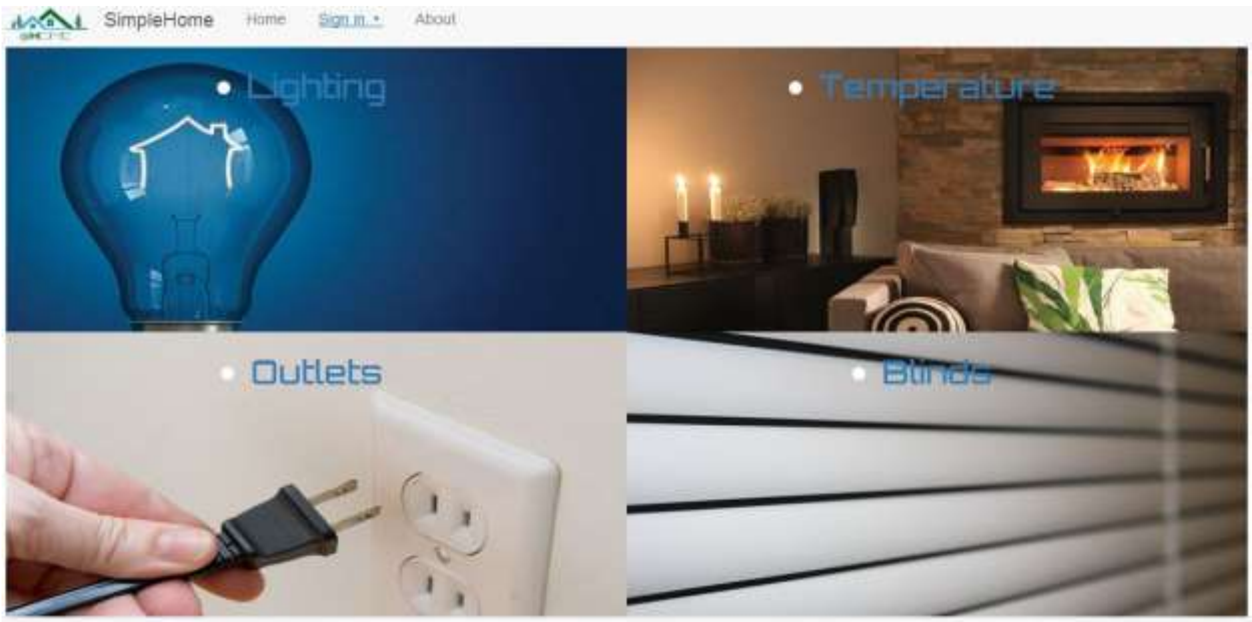

Figure 10: Home page with links to peripheral information

Upon selecting a web page for the desired peripheral, the user can monitor data collected from each peripheral through a scatter plot chart displaying the daily energy consumption. To create the chart, an array is obtained from the required field entries from each peripheral document in the database. The chart framework will then loop through the arrays to populate the graph. The figure below displays the chart layout for energy usage collected from the thermostat peripheral.

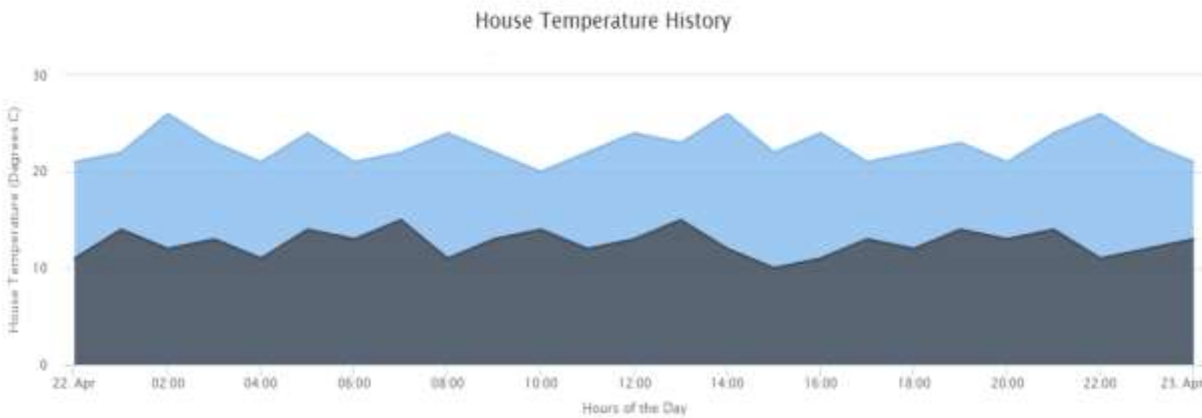

Figure 11: Scatter plot chart displaying energy usage from thermostat with mock data

The user will also have access to interactive buttons that will be located below the scatter plot chart to allow the user to control each peripheral. For switching the state of a peripheral, the button will hold the value 1 or 0 to represent “ON” or “OFF” respectively. When the button is pressed, the state field in the database will be updated and set to the value of the button. Similarly, the user can change the room temperature through the use of an adjustable knob button by selecting the desired temperature to update the temperature field in the database.

The internet database will bridge between the SimpleHome hub and the web interface. The database will store information collected from the peripherals and sort them into separate categories so that the information can be displayed on the website. In addition to storing data from the peripherals, it will also help to control the peripherals. Inputs changed on the web interface will as a result send info to the hub to change different states.

4.3. Data-Access Layer: Database Design

The backend that supports the website is based on a MongoDB database. The hub sends the data that it has collected from the SimpleHome Hub to the database. The database then organizes the data from the hub into the documents and fields so that the website can query the data and display it on the website. The figure below shows the hierarchy of the database in which the data from the peripherals are organized and stored in the database.

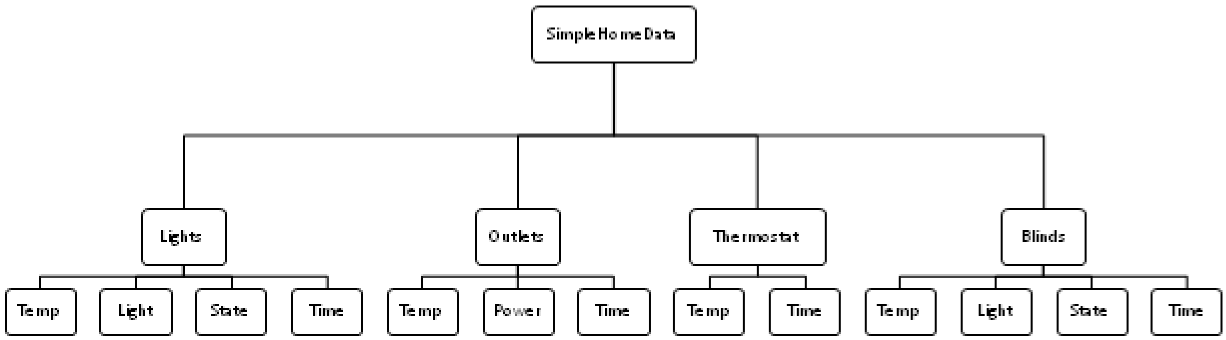

Figure 12: Database Hierarchy

A package that is added to the Meteor framework will handle the login and account aspect of the website. This package adds an accounts database in which the user's email address and password is stored. The hierarchy of this user database is shown below.

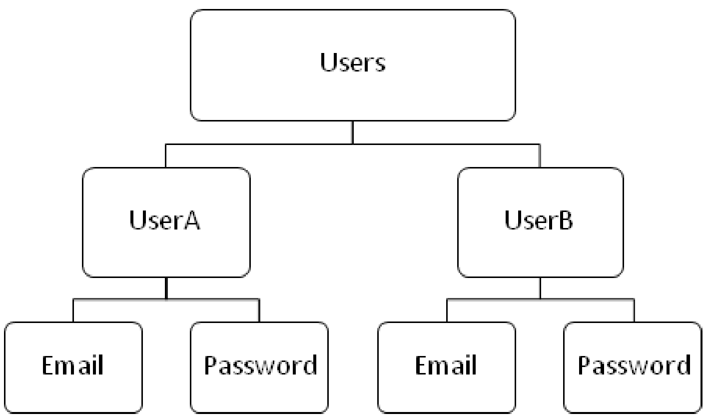

Figure 13: Hierarchy Design of User Login Database

Once you are on the main page, the user has to sign in in order to see the data from the database. The user has to create an account if they have not already done so and sign in. The flowchart below describes this accounts and login process.

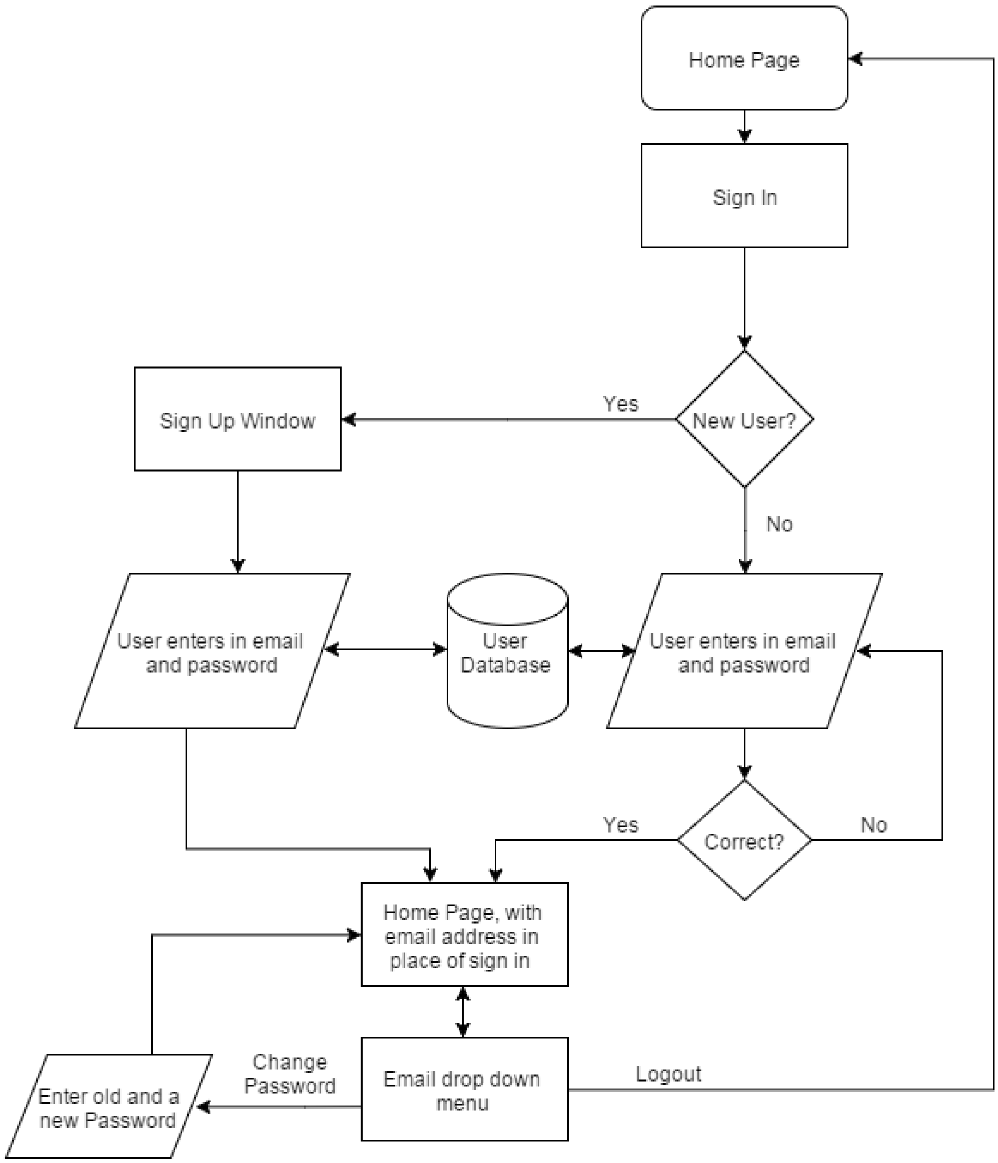

Figure 14: Flow Chart for Login Process

4.4. Further Development: App Design

The Meteor framework allows us to convert our meteor website into an app that can be accessible anywhere and by multiple mobile operating systems. Meteor uses a framework called Apache Cordova which wraps the website into a package in which most mobile devices can use. This process will make the user interface between the website and the app almost entirely the same making it easy for users to adapt and learn how to use. In addition, we would make an individually customized accessible app for each physically disabled and elderly individual for them to easily control the home automation system when needed.

5. Sustainability, Safety and Reliability

5.1. Sustainability

For the prototype model, the peripheral enclosures and the hub will be 3D printed using PLA filament. Although this material is not as durable as ABS, it is better for the environment as the plastic is made out of renewable starchy products such as corn starch, rather than being oil based [16]. As a result this plastic material is recyclable and biodegradable. When this product is produced for mass product, the hub and peripherals will be enclosed with HPPE plastic. Although this product is not as environmentally friendly as PLA, but this plastic is more durable and can take the heat from the sun making it a better option for our peripherals that will be outside. HPPE is not biodegradable, but can be melted back down to be made into to other plastic products.

To meet the requirement of well insulated wires, the wires that we use will be insulated with THWN insulation. This material is a thermoplastic insulation that is both, heat and moisture resistant while also being flexible as it is made of nylon making a great choice for our application. [17]

5.2. Safety

To prevent any accidental injuries the hub and peripheral enclosures are designed with rounded edges. Safety is a key factor in the selection of the components with most components being able to function under higher loads than the suggested. For example the wall plug using a SSR which can switch 25 A of current is used on a 10A circuit. All the electronics systems are designed to fail-safe where the system stays in its last state.

5.3. Reliability

The SimpleHome hub is designed to have a 10-year lifecycle for its components. The life of the Beagle Bone Black is projected to be the main source of failure in the SimpleHome hub and with a given lifespan of 100,000 power-on hours. In the case of the SimpleHome hub, it is assumed the system is always on so the expected life would be 11.4 years of continuous use that exceeds the desired 10-year lifecycle. [18]

6. User Documentation

For the initial demonstration of SimpleHome this is a low priority item as a physical document. Instead @HOME will focus on a visual skit that allows users to understand the interaction of the Hub and User Interface of SimpleHome with peripherals already in their own home. A short document may be prepared to explain functionalities and warnings of SimpleHome. In further developments, consultation with users will occur and a decision will be made on the most useful type of document to create, physical or electronic.

7. System Test Plan

The SimpleHome product design is very modular in nature. Therefore, unit testing can be carried out on each individual component first, and then the completed system can be tested as a whole. This test plan outlines the steps to be undertaken, in order to unit test the SimpleHome Hub, each individual peripheral and the user-interface. Once it is established that the all individual aspects of the system operate independently, we will follow the steps outlined in the overall system test procedure to verify the system's functionality. Upon completion of the complete system integration, regular use-case and extreme use-case scenarios will be tested.

7.1. Unit Testing

7.1.1. SimpleHome Hub

In order to test the Hub, a number of smaller tests need to be carried out, in order to ensure all functionalities are operational. Since the BeagleBone Black is the driving force behind the Hub, it will be tested individually, as well as, in sync with the Wi-Fi and ZigBee connectivity. The goal of these tests is to ensure that the Hub is capable of handling the high level of functionality which is expected from a home automation control system. Please note that these tests are manual, and a serial connection needs to be established to see the results of the connectivity based tests.

7.1.1.1. BeagleBone Black Test Plan

- Upon being connected to the power supply, the power LED should be on in a steady state. Additionally, if the OS is operational, there are four LEDs which should consistently keep blinking in a heartbeat fashion. This is an indicator that the OS was deployed correctly.

7.1.1.2. Manual Testing: ZigBee Connectivity

- Once the firmware is configured correctly, the LED on the CC2531 chip is on, in steady mode. In order to test the connection, we need to follow the steps outlined in the overall system design. ZigBee connectivity with devices cannot be tested unless there are ZigBee-enabled devices already available on the network.

7.1.1.3. Manual Testing: Wi-Fi Connectivity

Once the OS is operational and our Hub is ready, we will manually check the Wi-Fi connectivity using a two-step process:

- Upon using the 'ifconfig' command, a 'wlan0' connection should be displayed; specifying the details of the connection.
- Upon using the 'ping 8.8.8.8' command, the destination' response and time taken to respond will be printed on the terminal. The 8.8.8.8 IP links to the Google server. Additionally, we will also deploy the 'ping www.google.com' command--in order to check functioning with nameservers--and we should see a similar destination response.

7.1.2. Peripheral Device Testing

Each peripheral that uses our standard FW, which uses TCP commands to control the peripheral that means all our peripherals can be tested using the same procedure. The steps associated with the peripheral testing are listed below:

- The peripherals are listening on TCP port 43333
- Turn on the provided router
- Connect the peripherals TX, RX, and GND to a UART to USB converter
- Open a terminal program at 9600 baud rate
- Power up the peripheral, you should be able to see the IP address of the peripheral indicating a successful connection
- Start a new telnet session with the aforementioned IP address using port 43333
- Then send a command to see whether the peripheral responds, commands and responses are as follows:
- For switching peripherals, send SWITCHx=s where x (0-9) is applicable if there are multiple switching elements on that peripheral, and s (0 or 1) is for the desired switching state, no response over TCP is given
- For Data gathering peripherals, send GETTEMP for temperature, or GETLIGHT for light level, and you should receive a response back with the relevant data

7.1.3. User Interface

The test plan for the user interface will include the following test cases to ensure front and back end integration is functioning correctly:

- Observe that the user can log in with their credentials and view information from their peripherals
- Observe that the user is unable to access data if not logged in
- Ensure user can change password and logout
- Observe that the user can successfully change or vary the state of a peripheral with feedback from the button. This includes ensuring that updates made to the database from the front end are updated in MongoDB
- Ensure that peripheral data in the database will correctly update the energy usage graphs on the website on a daily basis
- Ensure that data is being stored in the database, by outputting data onto the server

7.2. Overall System Testing

7.2.1. Regular Use-Case Scenario

Since the peripherals send and receive information in conjunction with the user-interface the following tests will ensure the complete datapath is working. Since the operation of each peripheral is dependent on the it's Wi-Fi or ZigBee connectivity, testing the operation with one peripheral using each communication protocol implies that all the peripheral will operate, in essence, the same way. The following use-case being evaluated is for the Wi-Fi enabled automated blinds and the ZigBee-based third-party light bulb.

Task: To test the system sending/ receiving commands and functioning as intended for Wi-Fi-enabled devices, in particular the automatic blinds.
Action: 'The User' will send a command via the user interface to (1) open the blinds if they are closed, (2) close the blinds if they are open, and (3) do nothing if the blinds are in the same state as the user asks for.
Observations: This action must occur within 30 seconds of the user sending the command and take less than one minute to complete.

Task: To test the system sending/receiving commands and functioning as intended for ZigBee-enabled devices, in particular the Light Bulb.
Action: 'The User' will send a command via the user interface to (1) turn on the GE Link light bulb if it is off, (2) turn off the GE Link light bulb if it is turned on, and (3) do nothing if the light bulb is in a state that the user is requesting.
Observations: This action must occur within 30 seconds of the user sending the command.

8 Conclusion

The document outlines the design specification for SimpleHome's wireless home automation system. As such, the design specification highlights the company's goals to develop and ultimately manufacture a product targeted for an elderly person's use. The design specification along with the functional specification describes the functionality of our product and the implementation required to produce a working prototype. Additionally, the test plan details the approach we will take in order to ensure functional specifications are met.

The design specifications established will define our proof of concept and be used to verify the company's goals through the system test plan. The test sheets located in the Appendix will serve to measure the completeness and success of the system's functionality. As such, SimpleHome aims to deliver a complete and working prototype by December 14, 2015.

Appendices

A. Mechanical Drawings

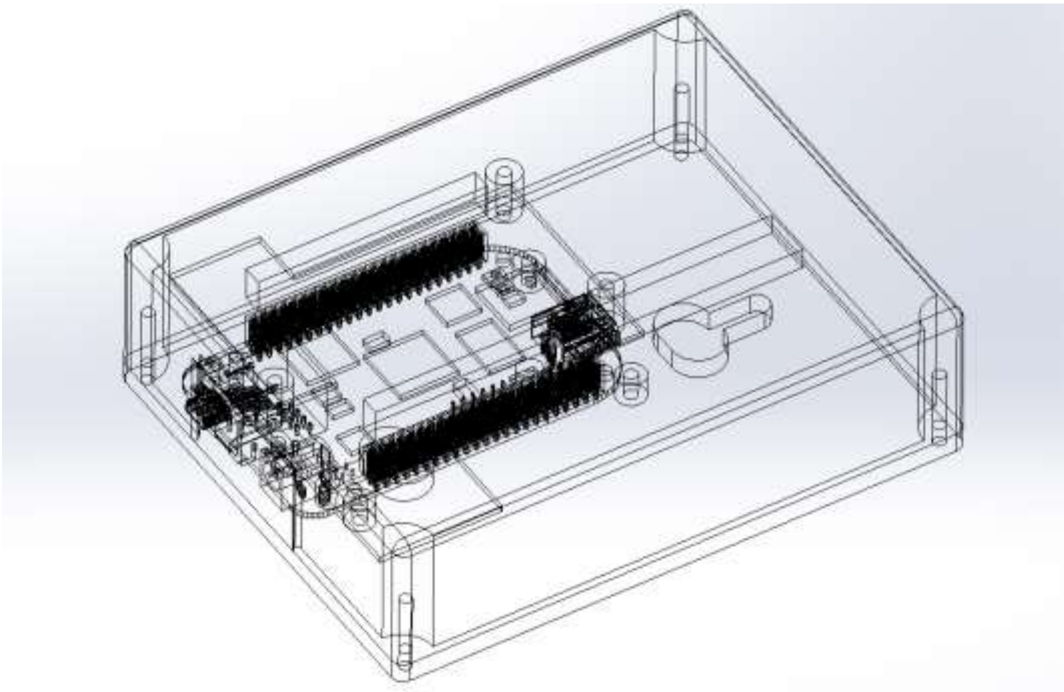

Figure 16: SimpleHome Hub Wireframe [13]

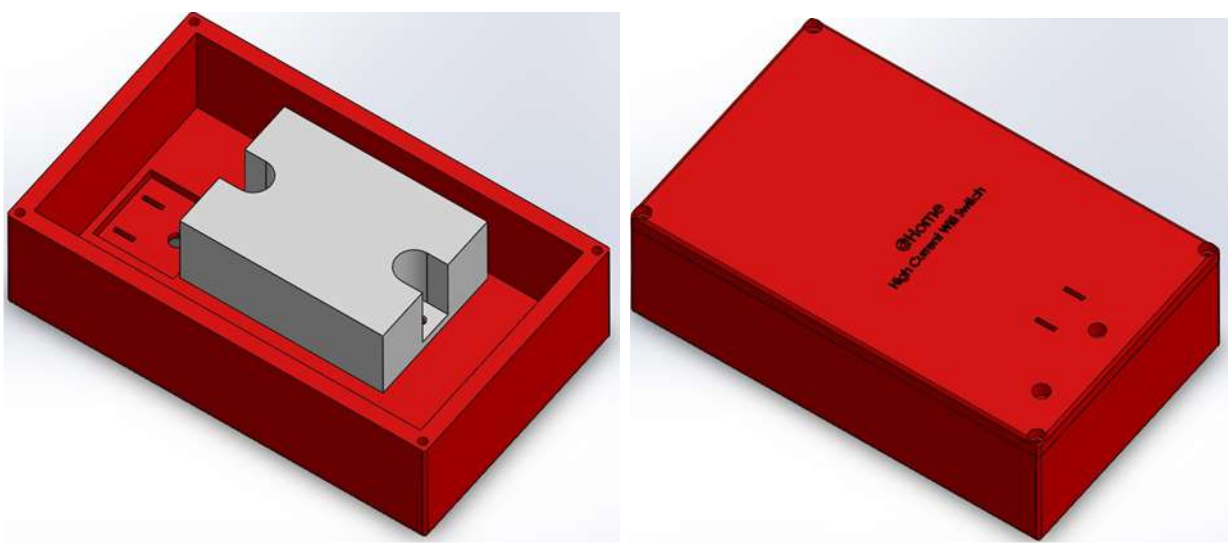

Figure 15: Peripheral box specifically for the wall plug enclosure

B. Electrical Schematics

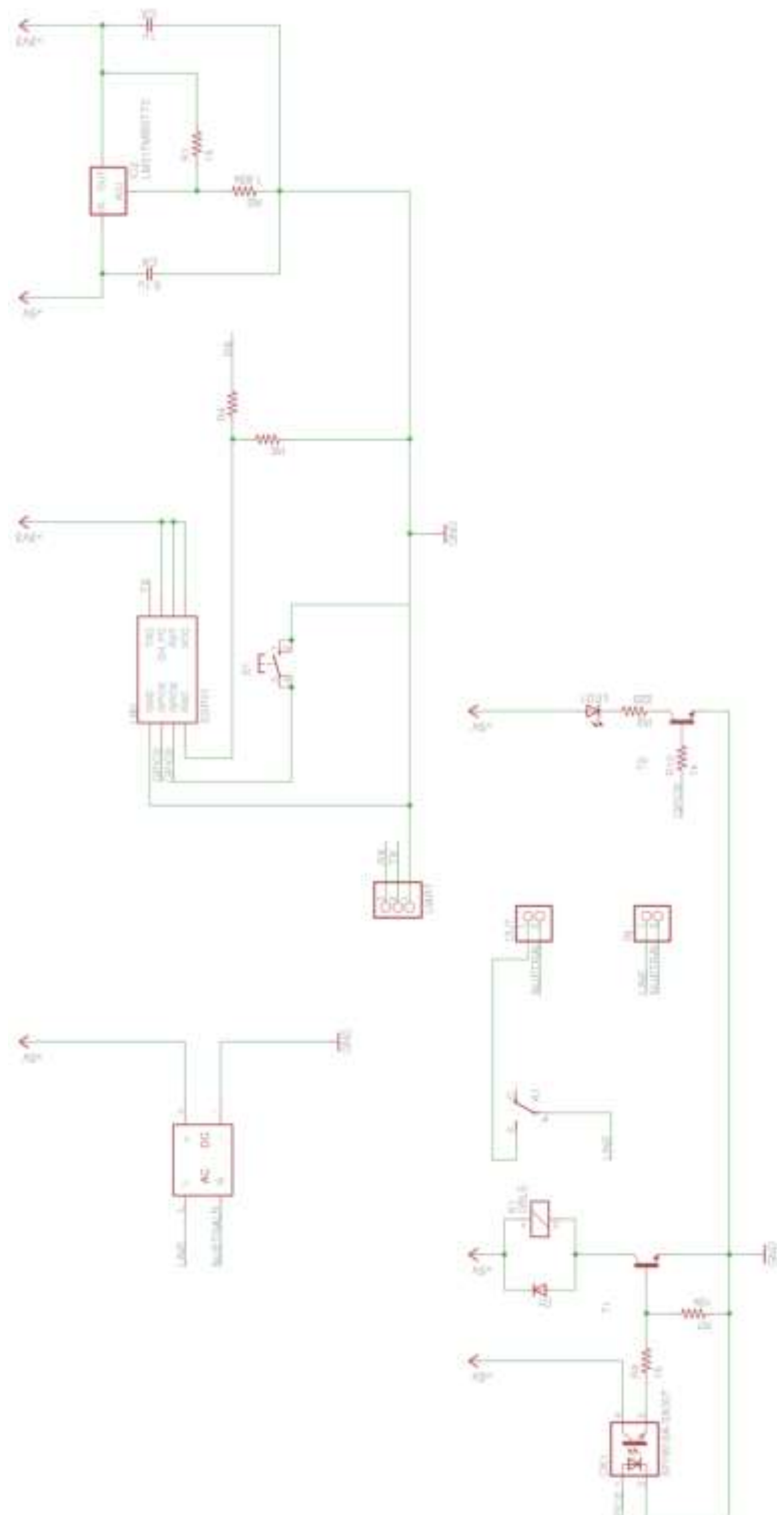

Figure 17: Switching circuit for light switch and electrical outlet peripherals

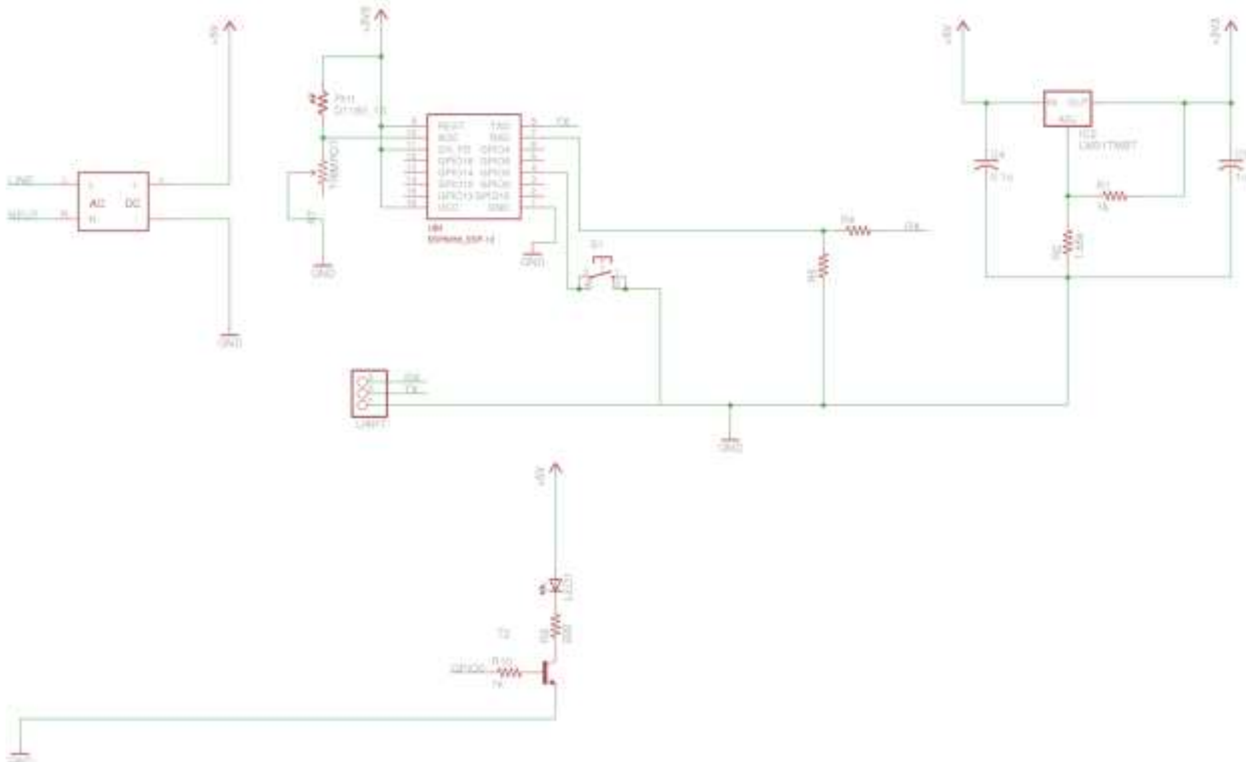

Figure 18: Light Sensor Schematic

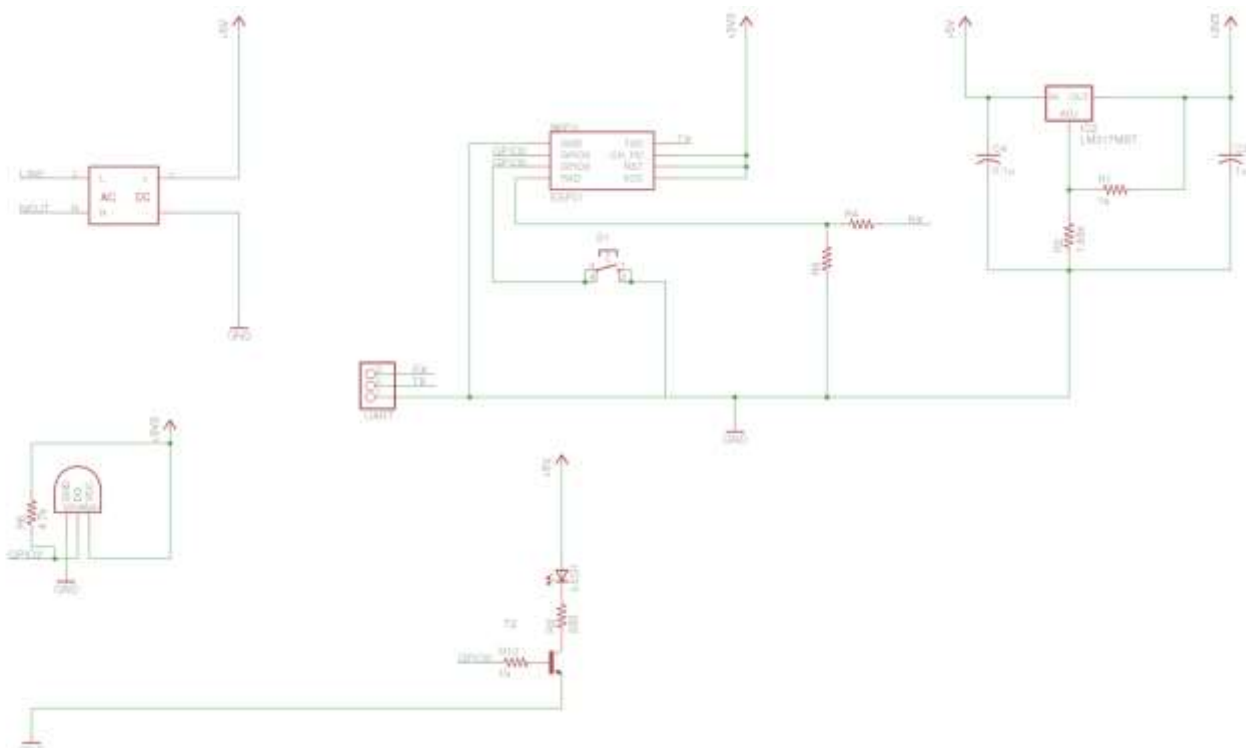

Figure 19: Temperature Sensor Schematic

C. References

[1] D. Spicer, "If You Can't Stand the Coding, Stay Out of the Kitchen: Three Chapters in the History of Home Automation," 12 09 2000. [Online]. Available: http://www.drdobbs.com/architecture-and-design/if-you-cant-stand-the-coding-stay-out-of/184404040. [Accessed 8 11 2015].

[2] T. Danova, "THE CONNECTED-HOME REPORT: Forecasts and growth trends for one of the top 'Internet of Things' markets," 2015. [Online]. Available: http://www.businessinsider.com/connected-home-forecasts-and-growth-2014-9. [Accessed 21 7 2015].

[3] Wikipedia, "Intel Edison," 10 June 2015. [Online]. Available: https://en.wikipedia.org/wiki/Intel_Edison. [Accessed 12 October 2015].

[4] webopedia, "MongoDB," webopedia, [Online]. Available: http://www.webopedia.com/TERM/M/mongodb.html. [Accessed 8 11 2015].

[5] Wikipedia, "Wikipedia," 30 September 2015. [Online]. Available: https://en.wikipedia.org/wiki/Window_blind. [Accessed 12 November 2015].

[6] WikiPedia, "Wi-Fi," 15 October 2015. [Online]. Available: https://en.wikipedia.org/wiki/Wi-Fi. [Accessed 17 October 2015].

[7] Wikipedia, "ZigBee," 11 October 2015. [Online]. Available: https://en.wikipedia.org/wiki/ZigBee. [Accessed 12 October 2015].

[8] "Government of Canada — Action for Seniors report - Seniors," 17 12 2012. [Online]. Available: http://www.seniors.gc.ca/eng/report/index.shtml. [Accessed 12 09 2015].

[9] "RoboSavvy," [Online]. Available: http://robosavvy.com/store/beaglebone-black-rev-c.html. [Accessed 12 11 2015].

[10] TI, "CC2531 USB Evaluation Module Kit," 31 7 2009. [Online]. Available: http://www.ti.com/tool/cc2531emk. [Accessed 8 11 2015].

[11] TI, "WL1835MOD," 11 2014. [Online]. Available: http://www.ti.com/product/WL1835MOD/datasheet. [Accessed 8 11 2015].

[12] BoardZoo.com, "BEAGLEBONE WL1835MOD W/ CHIP ANTENNA," BoardZoo.com, [Online]. Available: http://boardzoo.com/index.php/beaglebone/beaglebone-wl1835mod-w-chip-antenna.html#.VkSzHHarRhF. [Accessed 8 11 2015].

[13] Lgxlogic, "BeagleBoneBlack-3D-Model," Lgxlogic, [Online]. Available: https://github.com/lgxlogic/BeagleBoneBlack-3D-Model. [Accessed 6 11 2015].

[14] M. Grazier, "The future of home automation – ZigBee or Z-Wave?," 14 12 2013. [Online]. Available: http://www.edn.com/design/analog/4426050/The-future-of-home-automation---ZigBee-or-Z-Wave-. [Accessed 8 11 2015].

[15] S. Makonin, "2010 ZigBee Sensor Data Collected From a House (in Vancouver, Canada) Dataset," 2010. [Online]. Available: https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/2K9FFE.

[Accessed 08 11 2015].

[16] Eartheasy.com, "Plastics by the Numbers," 02 05 2012. [Online]. Available: http://learn.eartheasy.com/2012/05/plastics-by-the-numbers/. [Accessed 08 11 2015].

[17] Panduit, "Electrical Wire Sizes Selection Guide," 2011. [Online]. Available: http://www.panduit.com/heiler/SelectionGuides/WW-WASG03%20Electrical%20Wire%20Sizes-WEB%207-7-11.pdf. [Accessed 08 11 2015].

[18] Texas Instruments, "AM335x Reliability Considerations in PLC Applications," [Online]. Available: http://www.ti.com/lit/an/sprabv9/sprabv9.pdf. [Accessed 17 October 2015].

D. Test Sheet

Test Sheet	
Team: @Home	**Date:**

Unit Testing	
SimpleHome Hub • BeagleBone Black ☐ Power LED ☐ 4 LEDS flashing • Zigbee Connectivity ☐ Yes ☐ No • Wi-Fi Connectivity ☐ Yes ☐ No	Comments
Peripheral Device Testing • Light Switch ☐ Yes ☐ No • Power Outlet ☐ Yes ☐ No Temperature Sensor ☐ Yes ☐ No • Thermostat ☐ Yes ☐ No • Light Sensor ☐ Yes ☐ No • Automated Blinds ☐ Yes ☐ No	Comments
User Interface • No Access to data without Login ☐ Yes ☐ No • Login/Make new account ☐ Yes ☐ No • Logout and Change Password ☐ Yes ☐ No • Change state through Website ☐ Yes ☐ No • Data displayed on Graph ☐ Yes ☐ No • Data displayed on server ☐ Yes ☐ No	Comments

System Testing	
Regular Use • Light Switch ☐ Yes ☐ No • Power Outlet ☐ Yes ☐ No Temperature Sensor ☐ Yes ☐ No • Thermostat ☐ Yes ☐ No • Light Sensor ☐ Yes ☐ No • Automated Blinds ☐ Yes ☐ No • Zigbee Light Bulb ☐ Yes ☐ No	Comments