Senior
Thesis

An IoT Robotic Light Switch

Arsal Habib

Professor Steven Gustafson (Electrical Engineering)

and

Professor Doug Klein (Economics)

Submitted in partial fulfillment

of the requirements for

Honors in the Department of Electrical Engineering and Department of Economics.

UNION COLLEGE

Schenectady, New York

March 11th, 2016.

Summary

The purpose of this project was to design and build an Internet of Things (IoT) Robotic Light Switch. This light switch would act as a complementary device to existing toggle switches and equip them with typical smart device features such as remote control (via Wi-Fi), digital voice assistant control, motion-sensing and integration with other smart home solutions (such as IFTTT). All these features would be supported by a dedicated web page that would also be able to view the usage statistics of the device. This device is aimed at offering users an extremely easy-to-install home automation solution that needs no wiring and gives users the flexibility to implement a phased transition to home automation rather than having to revamp their entire home lighting system.

The device was designed to meet certain basic constraints. First and foremost, the device must be extremely simple to install to an existing light switch and should not require any professional services for installation. Secondly, the configuration of the device should be relatively simple and there should be ways to easily re-configure the remote communication method settings when moving the device from one location to another. Besides this, the device should offer a remote communication platform such as a webpage that can be used to control the switch, view all switching/sensor activity and integrate the switch with other smart home solutions/services.

The body of the device was 3D printed and an adhesive was used to snap the device on to existing switches. A webserver was set up to offer a web platform that allowed the use of all features of the device. Also, the webserver allowed integration with 3rd party services such as IFTTT and digital voice assistant. Integration with other services allowed additional features such automated switching based on motion-sensing and GPS-location of user, voice command control via Cortana/Siri etc. Finally, a business canvas model was developed using this device as a case-study. The model addressed all aspects that would need to be considered if the proposed device was commercialized including a detailed cost analysis.

Table of Contents

Table of Figures and Tables

Introduction

Each year an enormous amount of energy is wasted because of unused electrical appliances and lights left running. This is solely because of the unwillingness of people to make the effort and turn lights/appliances off when they are done using them. As we continue to move to a world where we make the most out of technology to make our lives easier, home automation has been seen as an answer to this problem. The idea behind home automation is that all lights and electric appliances in a building/house are equipped with the capability to turn ON or OFF automatically without a human having to monitor them. At the same time, home automation makes it easier for humans to control the appliances by providing remote access. However, the cost and effort involved in transitioning to a 'smart' home is unreasonable for most people. Present solutions are expensive and involve completely replacing existing systems which adds up to the complexity – acting as a huge barrier for those considering home automation. Additionally, a lot of people do not see it economically feasible to automate every appliance in their house or in the building and instead, want to only automate a few lights/appliances that they use frequently.

This capstone project proposes an Internet of Things (IoT) Robotic Light Switch that is easy to install on top of existing light switches. This robotic switch would therefore convert a conventional electric light switch into a 'smart' switch, giving it additional capabilities such as remote control, scheduling, motion-sensing, etc. Any electric appliance connected to the switch would hence be able to use all these automation features and act 'smart'. The device would be remotely connected to a web server that would allow users remote access to the appliance. At the same time, the web application would be able to automate the appliance based on numerous other factors such as temperature, light or motion sensors. All switching activity would be recorded and the data could be used to implement machine learning algorithms in the future. Upon completion of the project, this device would be installed to an existing wall switch and its capabilities would be tested against the requirements.

The remainder of the paper is organized into 8 sections. Section 1 provides an overview of the IoT and the home automation industry and, using surveys and studies, goes on to discuss the common barriers to home automation. The proposed device is tested against these barriers which are then used to devise the design requirements for the device in Section 2. Based on the requirements, Section 3 proposes a preliminary technical design to build a working prototype of the robotic switch. Section 4 examines alternative designs and provides justifications for functional parts and methodologies chosen. Section 5 presents the final design that was implemented to build a working prototype of the proposed device. Section 6 contains testing results and performance estimates based on the prototype while Section 7 provides a user manual for the device. Finally, a discussion of the results is presented in Section 8 along with further recommendations for the device.

In the Appendix A section, a detailed business model canvas is developed for the proposed IoT Robotic Light Switch. This model can be used to commercialize such a robotic switch and includes a detailed breakdown of expected costs if this device was mass produced.

Section 1: Background

I. Internet of Things (IoT)

Internet of Things (IoT) refers to a network where objects or living beings are given unique identifiers and are interconnected. They are able to transfer data over the network or the internet without requiring human-to-human or human-to-computer interaction. The unique identifiers are IP addresses which are allocated to each object in the network. The 'things' in IoT refers to non-living or living things. Examples could include patients that have some kind of a heart monitoring chip installed on them or a car that has a tracking sensor installed in it. Over the years, IoT has gained a lot more attention and has seen great progress because of the increase in address space on internet because of IPv6 (the latest version of the internet protocol). As of now IPv6 has enough address space to assign a unique IP address to every 'thing' on the Earth.

<u>Business prospects of IoT</u>

IoT has been constantly tipped to be one of the biggest revolutionary ideas over the last couple of decades but the industry has yet to see the kind of results predicted. McKinsey (Baruer, Paten and Veira, 2014) predicts that the IoT industry could be as big as $6.2 trillion by 2025 provided the corporate leaders effectively utilize the opportunity available. Similarly, Gartner estimates that the installed base of IoT devices (excluding PCs, tablets and smartphones) will grow to $26 billion devices by 2020 (Middleton, Kjeldsen and Tully, 2013).

II. The Home Automation Industry

One of the industries that has seen the greatest effect from the emergence of IoT is the home automation industry. Home automation refers to the use and control of home appliances remotely or automatically. The home automation industry is expected to grow at around 24.6% over the next 5 years and reach the

US$16.4 billion mark [1]. This growth has primarily been fueled by the growing need for effective solutions in various domestic applications such as lighting, energy management, entertainment etc. Recently the trend has started to shift to smartphone based home automation systems. Such systems give user the ability to monitor their household devices via apps on their cellphones from anywhere in the world. This allows for a much more secure household and substantially helps the energy consumption as well.

The home automation space has different dimensions to it which range from controlling kitchen appliances to security to lighting solutions to climate control devices. According to a study by Gale Group (Business Insights: Essentials, 2016), climate control solutions that help control heating and cooling lead the adoption curve while security, audio/visual and lighting solutions fall slightly behind. Table I below summarizes these results from the study by Gale Group that included 2600 participants who owned at least 1 smart home product.

Table 1: Adoption of various smart home categories based on survey (Business Insights, 2016)

Home Automation Category	**Percentage of participants who own a device in the category**
Climate Control (thermostat, air conditioning)	72%
Security (cameras, smart locks, smart alarms)	53%
Audio/visual (smart TVs and speakers)	52%
Lighting (smart lightbulbs, light switches)	46%

The study by Gale Group also indicated that among the people planning to buy a home automation device in the future, 24% were thinking of buying a smart lighting device, 25% of a security device and 12% of a resource management device.

Despite the appeal of smart home technologies, home automation, just like IoT, has also failed to achieve the hype and the market estimates that were set for the industry a few years back. Such concerns have recently resurfaced after some of the most anticipated home automation startups declared bankruptcy

(Tollentino, 2013). In the following section, some of the reasons for this slowdown in growth are mentioned and discussed in detail.

Barriers to Home Automation

There have been various studies carried out to analyze the adoption of home automation technologies over the last couple of years. These studies are aimed at highlighting the main issues with present technologies so that future products and services can try to fix them. Brush (2011) and Bannister (2015) pointed out these barriers after surveying groups of consumers who had either implemented smart home technologies or were planning to do so. These barriers include high cost of ownership, inflexibility, security and lack of usefulness of devices.

<u>Cost of Ownership/Installation:</u>

The cost of ownership can either comprise the monetary costs or the time-related costs in the implementation of these home automation technologies. One of the major considerations in the cost aspect is whether the product is do-it-yourself (DIY) or a complete installed solution (outsourced). However, as Brush (2011) found out in his survey, the cost of hardware alone could be very expensive at times. Often, individual panels to replace standard light switches cost around $100. Not only did most of the households end up paying for the hardware, they also needed assistance from home automation consultants when transitioning to a smart home. This quickly adds up the costs as consultants and technicians who install these systems can be quite expensive.

A major focus in home automation is automating the lights in a house. According to the U.S. Energy Information Administration, 15% of total US residential electricity is used for lightning purposes [2]. Much of this energy is wasted because of unused lights and appliances left running. Therefore, this is one of the easiest ways for households to control their utility bills which have skyrocketed to a national average of $1369 over the last decade [3].

On average, an American household has around 21 lights each controlled by a light switch [4]. These light switches are mostly toggle style switches while some households use the more modern form, the "rocker" or the "paddle" style. Replacing these with switches that could allow home automation and could be controlled via a cellphone app would take a minimum of $1000 for an average household and the labor costs will further inflate that figure. A typical example of such a switch is a WeMo Light Switch which sells for $49.99 and has to be installed by a certified electrician. According to Fixr (a business that connects professionals to customers), the average cost for licensed services of an electrician is $85/hour. Hence, the cost and the effort required in upgrading to a smart home is enormous and the benefits in the form of ease of access and energy savings, might not be enough for a typical household to justify the transition.

<u>Inflexibility or lack of interoperability</u>

Inflexibility of home automation devices refers to the failure of present solutions to communicate and integrate with home automation devices from other companies. While there is no shortage of products in the home automation industry being revealed every other week, consumers often complain about the lack of flexibility of these devices. Since the home automation industry is relatively new, standards have not yet been set. Because of this, every player in the market utilizes different technologies which are often not compatible or operable with other products from competitors. And since the home automation space has various dimensions to it ranging from lighting to security to climate-control solutions, it is not possible for a single player to provide solutions in each space. This leaves the consumers in a fix when they are trying to decide what products to choose and how to get them to work with each other. Not to mention that some companies intentionally lock consumers so that they can only integrate their devices with other devices from the same company. This ultimately serves as a huge barrier to adoption of home automation and calls upon the present/future products to devise architectures that allow them to interoperate with various other devices.

Lack of purpose of devices

Findings by Brush (2011) indicate that a lot of people perceive a really low value to a lot of the benefits that home automation solutions provide. This generally refers to devices that are equipped with extra features that are simply not solving any problem the consumer might be facing. Lots of these features are instead making consumers' lives even more complicated rather than simplifying them. An example of this added complication is sometimes the requirement for users to understand complex algorithms that smart devices are running in order to understand the behavior of the device. Secondly, most home automation solutions come with a complete package and hence the consumers often need to revamp the entire existing system in their homes when they adopt a certain home automation system/product. Not only does this increase the cost, but also complicates the process and the user ends up owning more devices than needed to actually solve the intended problem.

III. Proposed Solution: An IoT Robotic Light Switch

In light of the above mentioned statistics and surveys that highlight the concerns most buyers of smart home technologies have, this thesis proposes an **IoT Robotic Light Switch**. The viability of such a robotic light switch will be studied in this paper and a business plan will be discussed in detail to roll out this light switch commercially. Careful consideration was made during the design of the proposed light switch in order to address the concerns and barriers discussed earlier in the thesis that home automation technologies often face. This robotic light switch will be an add-on device for traditional rocker light switches that would equip them with smart features such as remote control (via Wi-Fi), digital voice assistant control, activity tracking, and motion sensing. Below, some features of the proposed robotic switch are discussed and special emphasis is placed on how they will address some of the barriers that were earlier mentioned in the paper:

Installation of device:

The robotic light switch will need no wiring, thus avoiding the need to contact a technician during the installation stage. It will have an adhesive on the back side that will allow the user to simply attach the device on top of an existing light switch in little time. The device will then actuate the toggle switch ON or OFF by providing the required mechanical force. In this manner, the installation of this device would address the concern that most prospective buyers of home automation technologies have in terms of the costs associated with installing smart home solutions.

Transitioning to a smart home (phased implementation):

Another feature that this robotic switch provides that is often missing in other lighting solutions is that it will allow the users to pick the lights in their houses that they want to automate. While most other lighting solutions force the users to revamp the entire lighting system in their houses, the proposed solution will give the users an option to implement a phased transition to smart homes leaving them with the freedom to pick the lights that they want to automate. Users will have the option of starting out by just automating a single light in their houses. Because of the simple installation process, users would also be able to move around this device from one light to another based on their needs, thereby giving them a lot of flexibility.

Integration with other smart home solutions:

The proposed switch has been designed with the interoperability aspect in mind. Hence, the design has been open-sourced to encourage its integration with other smart home solutions and services such as IFTTT[1]. Also, the web server complementing the device gives users access to switching and motion data recorded by the device which can be used with other services and devices. For example, the motion sensor data from the device could be used with other smart devices that may be present in the same room.

[1] IFTTT (If this then that) is an IoT service that allows users to automate events based on triggers.

Hence, the proposed light switch also addresses the concern that consumers have of smart home devices not being interoperable with each other.

IV. Security Issues

Security issues pose a huge threat to not only the proposed device, but the whole home automation industry in general. Since most home automation systems and devices rely on the internet or on some kind of a wireless network, there is always the threat that hackers might get into the system and steal important information that could expose the security of homeowners.

Such an example of a security threat involving the proposed actuator could be if some outsider hacks into the web application and steals data about the switching activity. The switching activity would then reveal timings of when the user is present in the house/room or not. This could pose a huge security threat for the user.

During the design of the actuator device, basic security features were kept in mind. A unique ID would be assigned to each actuator and a password would be needed to access the actuator. Also, a reset method was incorporated into the design that would wipe out the recorded data if need be. For the purposes of this project, it was decided that rather than spending too much time on providing robust security for the device network, it would be more fruitful to spend more time working on additional features of the device. This approach was not meant to undermine the threat of security but only chosen because of the limited amount of time at hand. As mentioned above, basic security features were still included in the design.

V. Product Sustainability

The actuator device has a lot of room for further development and would have to evolve as more progress is made in the home automation industry. During the design process, it was ensured that the device could adapt in the future. The data recorded by the actuator is an example of that adaptability. This data could, in the future, be used to implement analytics and pattern learning algorithms that would help the actuator predict usage.

Similarly, the proposed version of the actuator device would only be compatible with toggle type switches. In the future, the design could be altered to allow rocker-style switches which are universal in Europe and Asia. This would open up a huge market overseas since toggle switches are only common in the North America region.

Another aspect of the actuator device that could be explored is the additional features one. The actuator could be customized to provide various additional features such as motion sensors or light sensors. These sensors would help users monitor their rooms or houses even when they are away and hence the actuator device could also help in terms of security.

Section 2: Design Requirements

In this section, the desired behavior of the proposed device upon completion will be described. This behavior will provide the ground rules for discussion of the design alternatives and the final design in the next two sections. The final prototype must meet or exceed the functional and non-functional design requirements presented below.

I. Functional Requirements

<u>Dimensions</u>

The switch device should fit a standard switch plate which is 7cm x 11cm (single gang). Additionally, it should be small enough such that multiple devices can be attached to a wall-plate that has multiple gangs/switches. A standard 2-gang switch plate is shown below in Figure 1. In order to install 2 switch devices on such a wall plate (one for each toggle switch), the device should have a width of no more than 4.5cm and the length should be less than 10 cm.

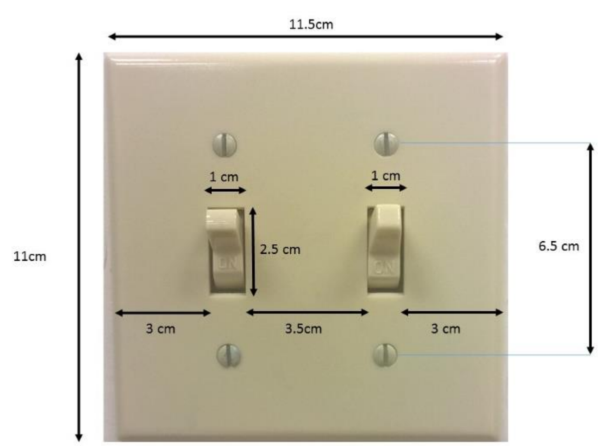


Figure 1: Dimensions of a standard 2-gang switch plate.

Attaching actuator on to plate/switch

The installation of the device on to a toggle switch should be extremely simple and should not need any extra tools or help from a technician. Upon removing the actuator from the switch, the switch plate should not be damaged or tampered with. The method of attachment should also be strong enough so that the device does not fall off the plate or slide against the plate when the button is flipped.

Mechanical Action

The actuator device should be able to provide the mechanical action to push a toggle flip switch ON or OFF. Standard toggle switches were tested and it was observed that they needed a force of around 6N to be successfully flipped. Hence the actuator should be able to provide a force of at least 8N so that it does not stall in the process.

Remote Connection

Since the desired switch actuator should act as a smart device, there should be a method to remotely communicate with the device. The communication method should be readily available such that any person with a smartphone can connect to it. Also, the chosen communication method should have an infinite range such that the user can connect to it from anywhere. When the actuator is used for the first time, the user should be able to set the settings for the communication mode. There should be a reset button on the actuator that resets all these settings.

Webpage and smartphone application

There should be a webpage that allows the user to control the actuator device. This webpage should support all additional features in the device. It should show the recent activity of the actuator and of the motion sensor along with a timestamp and should also continuously update the present state of the switch. The webpage should ask for the device ID and a password before allowing the user all these features.

<u>Location-based switching via smartphone app</u>

Additionally, a smartphone app should allow a feature such that the actuator switches ON when the smartphone is within 25 meters of the actuator. Similarly, it should also provide the option that the actuator switches OFF when the smartphone is more than 25 meters away from the actuator. The user should be able to switch both these features ON or OFF from the smartphone app. Figure 2 below shows a block diagram to understand the connection between the smartphone and the device.

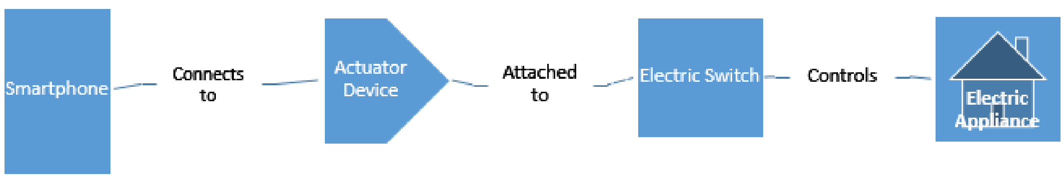


Figure 2: Block diagram showing connection between smartphone and device.

<u>Switching based on motion sensor</u>

The actuator should have an added feature that allows it to detect any movement or motion around it. The smartphone app and the webpage should allow the user to select when this feature is activated and when it is not. Additionally, the smartphone app should show the last time the motion sensor detected any kind of motion around it.

II. Non-Functional Requirements

Besides the functional requirements listed above, the system must also meet a number of non-functional requirements. The fundamental value proposition of a device such as the actuator being designed is that it is a cheap alternative for people looking at home automation and it is extremely easy to install without requiring any professional services. Hence the actuator device must satisfy both these requirements. It must be an affordable device and the components selected during the design stage must satisfy that criteria. Secondly, it should be designed such that anyone could install and configure it without needing any professional services. This requirement should be especially kept in mind while considering design alternatives for attaching the device on to an existing switch and during the configuration stage of the actuator. The web and smartphone application should be designed in such a way that they can accommodate multiple devices.

III. Test Strategy

Upon completion of the design, the device will be tested to ensure that it meets the non-functional and functional requirements mentioned above. Table I below shows how each feature will be tested.

Table 2: Testing Strategy and Criteria.

Feature Being Tested	Test Strategy	Test Criteria
Device Size	Two devices will be installed on to a switch plate with 2 gangs.	Both devices should be able to fit two switches easily.
Device Installation	Device will be installed on to a switch and later removed.	Connection should be strong. No damage to switch/plate.
Mechanical Action	ON and OFF commands will be given to the actuator via app 5 times.	The switching action should be smooth without any stalling.
Remote Connection	Actuator will be configured and commands will be input remotely.	Communication method should work after configuration.
Web Application	Device will be used via the web app.	ON/OFF and recent activity features should work.
Proximity Switching	Feature will be tested by walking towards and away from device.	Should turn ON/OFF when 20m near/away device.
Motion Sensor	Movement provided around device to check whether it reacts.	Switch should turn ON when movement detected.

Section 3: Preliminary Proposed Design

I. Final Proposed Design

Figure 6 below shows how the components inside the actuator device will be arranged. The top part of the device will hold the required batteries to power the device. Towards the middle of the device, there will be a switch slider that will hold the switch button. This slider will be connected to a servo motor through an arm. There will also be a top cover for the device that will have an LED showing the connection status of the device. The device will be attached to the switch plate via an adhesive strip. The remaining parts in this section will further detail the design choices.

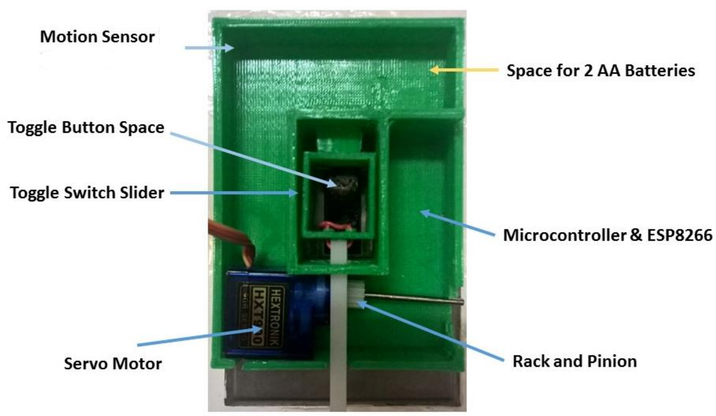


Figure 3: Final schematic of actuator device.

II. Device Case

The housing for the actuator device will be 3D printed and will have a width of 4.5cm and a length of 8cm. In the middle of the device, there will be an arm inside which the toggle switch will fit. The arm will be connected to a servo that will control its position. The user will attach this device on to an existing toggle switch using an adhesive strip provided with the device. The adhesive strip will be strong enough to hold the device on to the plate. Also, the adhesive strip will not destroy the switch once the device is removed.

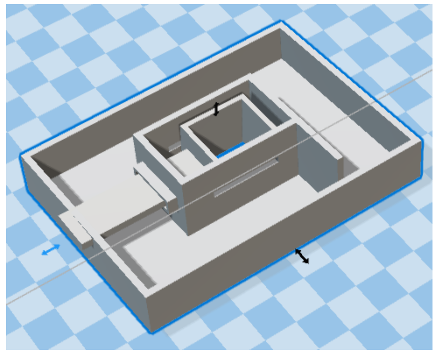


Figure 4: 3D design of actuator housing.

III. Mechanical Action

The mechanical action to flip a toggle switch ON/OFF would be provided by a servo. The servo will be connected to an arm that controls the toggle switch as shown above in Figure 3. Since a force of around 8N is needed to flip a switch, HXT900 micro servo will be used [9]. HXT900 micro servo provides a torque of 1.6kg-cm that is more than enough to exert a force of 8N on the toggle switch via the arm. The voltage rating of this servo is 3-6V and hence will work with the 3.3V supply needed for the ESP8266 microcontroller. Since a servo only provides rotational movement, a mechanism will be needed to convert it to linear motion that can flip a toggle switch ON/OFF. A simple rack and pinion mechanism will be used to convert the servo motion to linear motion. Figure 4 below shows how the rack and pinion mechanism will be connected to the servo and provide linear motion to the connected arm. The pinion is directly connected to the arm that will flip the toggle switch. 2 AA batteries will be used to power the switch actuator providing around 3.2V which is sufficient for all the components used.

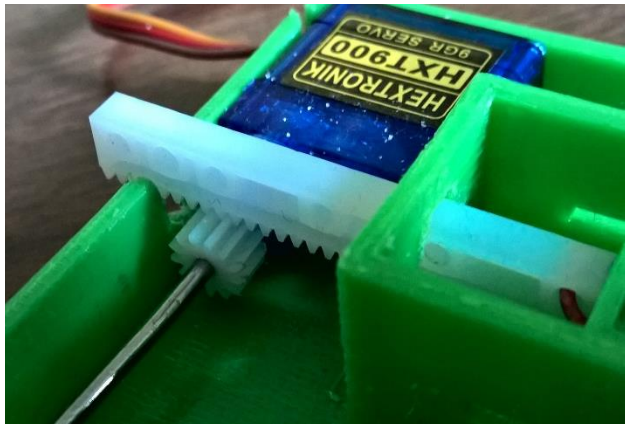


Figure 5: Rack and pinion mechanism used to convert rotational movement to linear motion.

IV. Wi-Fi Configuration Mode

For the actuator to connect to a Wi-Fi network, the local Wi-Fi credentials will have to be first entered into the ESP8266’s memory. This will be done by entering configuration mode by pressing a reset button on the actuator. In configuration mode, the actuator device will act in access point mode and transmit its own Wi-Fi. The user will be able to connect to this network through their smartphone or their laptop. Once connected, the user will be redirected to a webserver that will list down all the Wi-Fi networks that are within range of the actuator device. The user will be prompted to choose his/her local Wi-Fi network and enter its credentials. After the credentials have been entered, the actuator will try to connect to that Wi-Fi network. If the actuator is successfully able to connect to the local Wi-Fi, it will save the Wi-Fi credentials for future use and disconnect from access point mode and come out of configuration mode. An LED on the device will indicate whether the actuator is connected to Wi-Fi or not.

V. Network Topology

A web folder on a server will act as the focal point of the network design. All devices including the smartphone app, webpage and the actuator will directly post to and access data from the web folder via Wi-Fi or internet. Figure 5 below shows an overview how such a network would work. The mobile application and the web page would submit all requests to the web server via Wi-Fi. The mobile app and the web page would also have read access to some data from the web server (recent activity, motion sensor data). All requests would be submitted to the smart switch device via the web server. The actuator device would post data such as motion sensor readings and switch activity to the web server. The website and the smartphone application would be able to read all this from the web server and display it to the user. In order to change the present status of the actuator, the application or the webpage will have to submit a request to the webserver which will then be read by the actuator.

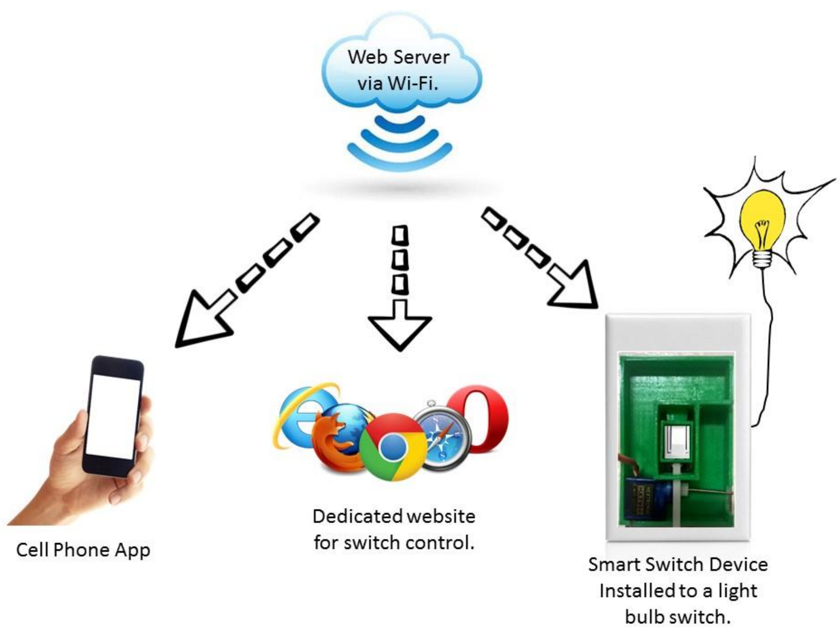


Figure 6: Network Topology

VI. Location-based switching via smartphone app

The smartphone app will also provide location-based switching features to the user. The two features will be:

1. Actuator switches ON when smartphone is less than 25 meters from the device.
2. Actuator switches OFF when smartphone is more than 25 meters from the device.

In order to implement this feature, the smartphone app will use the smartphone's GPS to detect its location and then compare it with the device's position. The smartphone will post its location on to the webserver. The actuator will read that location from the webserver and perform the required action after comparing the location with its own location. Since the actuator device is not equipped with a GPS module, it will determine its location during the configuration stage using the configuring device's (smartphone) GPS and save it for future use. If the device is moved to a different location after configuration, it will have to be re-configured. Indoor GPS can generally provide coordinates with an accuracy of around 17m, however, it will vary depending on the building dynamics [11].

VII. Motion Sensor

The actuator device will be equipped with a motion sensor. If the user has selected the motion sensor feature from the device's settings in the smartphone application, the application will display the most recent time at which motion was detected around the switch. Also, the switch will turn ON whenever motion is detected around the switch. The motion sensor used in this device will be a passive infrared sensor (PIR) that works by measuring infrared light radiating from objects in its field of view. This sensor will need around 3-3.3V which will be provided by the same power supply the microcontroller is attached to.

Section 4: Design Alternatives

I. Remote Connection Method

The remote connection method is very essential to the device because of its nature and the effect it has on a lot of the other design choices. There were two serious options considered for the remote connection method for this project. The two technologies considered were Bluetooth and Wi-Fi (802.11 wireless networking standards).

Bluetooth Connection is a wireless technology standard for exchanging data over short distances. Hence, if the actuator was to use Bluetooth to connect remotely, it would have a very limited reach. Only users within range would be able to use the actuator device which means that there would be no way to even see the status of the actuator, let alone control it, when not in range or outside the house the device is installed in. Also, the user would always have to keep their smartphone's Bluetooth ON to communicate with the switch actuator device. Potential customer interviews have indicated that this is not something desirable since it drains out the phone's battery faster. Another issue with using Bluetooth as the primary mode of remote communication is that it would not be able to support a webpage and would only work with a smartphone's application. This would restrict the usability of the device to only those who wanted to use their smartphone to control the device. Therefore, because of the limited reach of a Bluetooth based connection and the user concerns regarding mobile battery drainage, it was decided that Bluetooth will not be used for this device as it does not satisfy the design requirements.

Wi-Fi technology is based on local area wireless computer networks. Electronic devices can connect to the network that is typically created by a routers or an access point. If the actuator design is based on Wi-Fi communication method, the switch actuator device would essentially have an infinite reach. This means that users would be able to control and monitor the switch activity from anywhere in the world as far as there is an internet connection available. A study by Strategy Analytics in 2012 revealed that 61%

households in the US have a Wi-Fi connection through a wireless router [6]. Furthermore, Pew Research found that 64% of the US adults own a smartphone [7]. With 61% households having Wi-Fi access and 64% adults owning smartphones, it would be reasonable to base the design of the switch actuator on Wi-Fi networks. Also, most smartphone owners already have Wi-Fi or data switched ON at all times. Hence unlike a Bluetooth-based design, a Wi-Fi based one would not result in any additional battery being consumed by the user's smartphone. Therefore, because of these advantages of Wi-Fi connection over a Bluetooth connection, it was decided that the actuator design would be based on Wi-Fi connection.

II. Hardware Alternatives

Microcontroller and Wi-Fi Module

Besides a basic microcontroller to power the servo motor and the sensors inside the actuator device, a Wi-Fi module is also needed. Wi-Fi modules are extensions to microcontrollers that allow internet connectivity. Two approaches were considered for providing Wi-Fi connectivity to the actuator device.

The first and the more conventional approach considered was using an Arduino microcontroller along with an Xbee Wi-Fi module. The Arduino microcontroller in this case would be the Arduino Pro Mini because of its small size (0.7"x1.3") as compared to the other microcontrollers. However, the combined size of the Xbee Wi-Fi module and the Arduino Pro Mini was still greater than the desired device size. Secondly, the Xbee Wi-Fi module cannot act as an access point which is essential during the configuration stage of the actuator. When acting in access point mode, a module is able to create its own Wi-Fi network which other devices can then connect to. During the configuration of the actuator device, access point mode is essential so that the user can communicate with the actuator and insert local Wi-Fi details which the actuator can then connect to for future use. Hence, the Arduino-Xbee approach was rejected since it would violate the dimensions and configuration mode requirements of the actuator device.

The second approach considered for adding Wi-Fi connectivity to the device was using and ESP8266 Microcontroller [10]. ESP8266 is an Internet of Things (IoT) device that can either be used with another microcontroller to act as a Wi-Fi module or it can be used as the sole microcontroller while providing Wi-Fi connectivity. The ESP8266 chip also has the capability to act in access point mode which allows it to create its own Wi-Fi network. The ESP8266 has dimensions of 2.4cmx1.6cm which satisfy the dimension requirements. Since the ESP8266 has the capability to act as a microcontroller while providing enough power for the servo, it could be used without any other microcontroller for the purposes of this project. Additionally, the ESP8266 costs less than the Arduino approach. Table 1 below shows a cost comparison for both these approaches.

Table 3: Cost comparison of Arduino-Xbee vs ESP8266 Approach. (Quotes from sparkfun.com)

Approach	Components	Cost	Total Cost
Approach# 1	Arduino Pro Mini Xbee Wi-Fi Module	\$9.95 \$43.95	**\$53.90**
Approach# 2	ESP8266-12	\$6.95	**\$6.95**

Based on the above analysis, ESP8266 not only satisfies the functional design requirements of the actuator device, but also costs less than the Arduino approach. Hence, it was decided that ESP8266 will be used for this project to act as a microcontroller and as the Wi-Fi module.

III. Software Alternatives

Network Design Alternatives

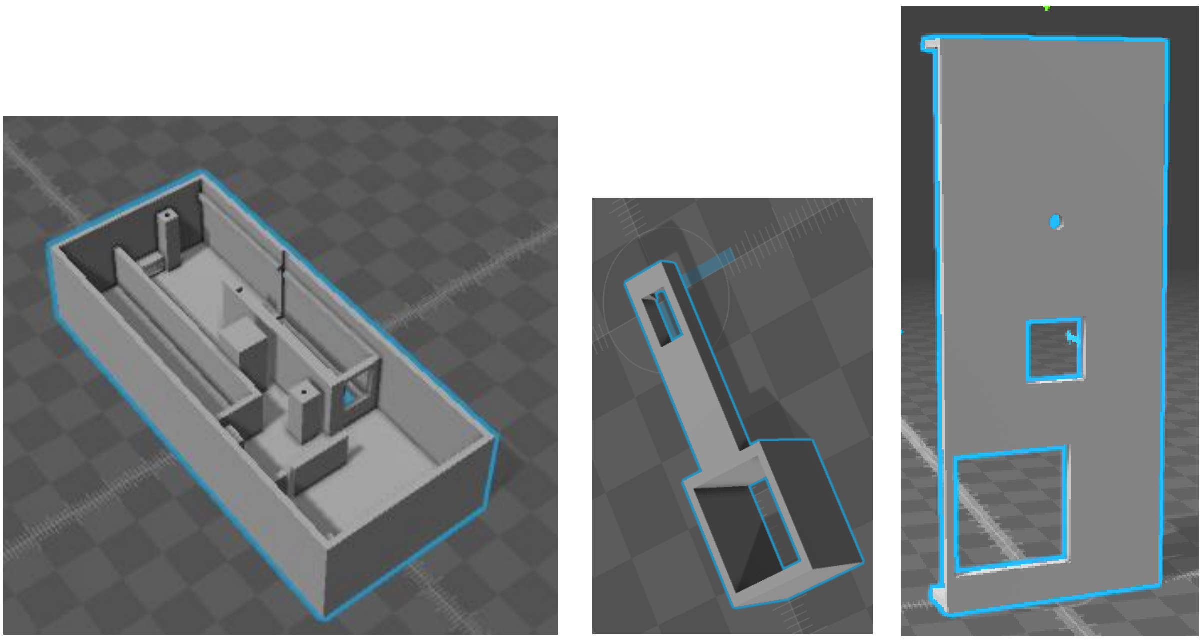
There were two network design methods considered during the design process. The first method involved storing all the data on the microcontroller inside the actuator device. This way the user’s smartphone app or the web page would have to communicate with the device to access any data. Because of complete dependence on the actuator device in this method, any breakdown in communication between the actuator and the web would disrupt all features of the webpage and of the smartphone application. This could potentially also result in all data being lost in case something happened to the device. Another concern with this method is that the amount of data that can be stored on the microcontroller is limited because of its memory.

The second method considered for network design involved storing all data on a web folder. This approach would require all devices including the user's smartphone, device webpage and the actuator device to connect directly to the web folder via Wi-Fi. All data from the sensors in the actuator would be posted and stored in the web folder. This way the smartphone's app or the webpage would be able to access data from the web folder even if there's no connection between the web folder and the actuator device. This method also allows more data to be stored since the web folder would typically have much larger storage capacity than the microcontroller. Because of the higher stability and more storage space, it was decided that the second method (store data on web folder) will be used for the actuator device design.

Section 5: Final Design and Implementation

I. 3D Design

The body of the device was 3D printed using Makerbot Replicator 2[2]. The 3D model was created in SolidWorks and comprised of 3 main parts: the main body, the pusher and the cover. Figure 7 below shows drawings of these three parts:



(a) Main Device Body (b) Pusher Inside Device (c) Device Cover

Figure 7: 3D Models of all three parts of device (Main body, pusher and cover)

The main body part in Figure 7(a) is the part that hold all components inside the device. This is also the part that is attached on to the light switch or the wall plate. The pusher in Figure 7(b) is installed into the main device body separately. One side of the pusher is latched on to the light switch while the other is connected to a servo. Hence, when the servo moves, it pulls the pusher which also pulls the toggle switch with it. This is how the mechanical action is achieved in order to control the toggle switch. Finally, Figure

[2] 3D Printer at Union College Design Studio

7(c) shows the cover of the device. The cover goes on top of the device once it is installed on a light switch. The cover has holes in it to allow for the override button, LED and the motion sensor to fit properly such that they can be accessed once the cover is placed on top of the device. Figure 8 below shows a printed version of the main body device along with the pusher fitted inside it. Various support pillars inside the device to attach and support components such as the motion sensor, push button and the ESP8266 are labelled in Figure 8:

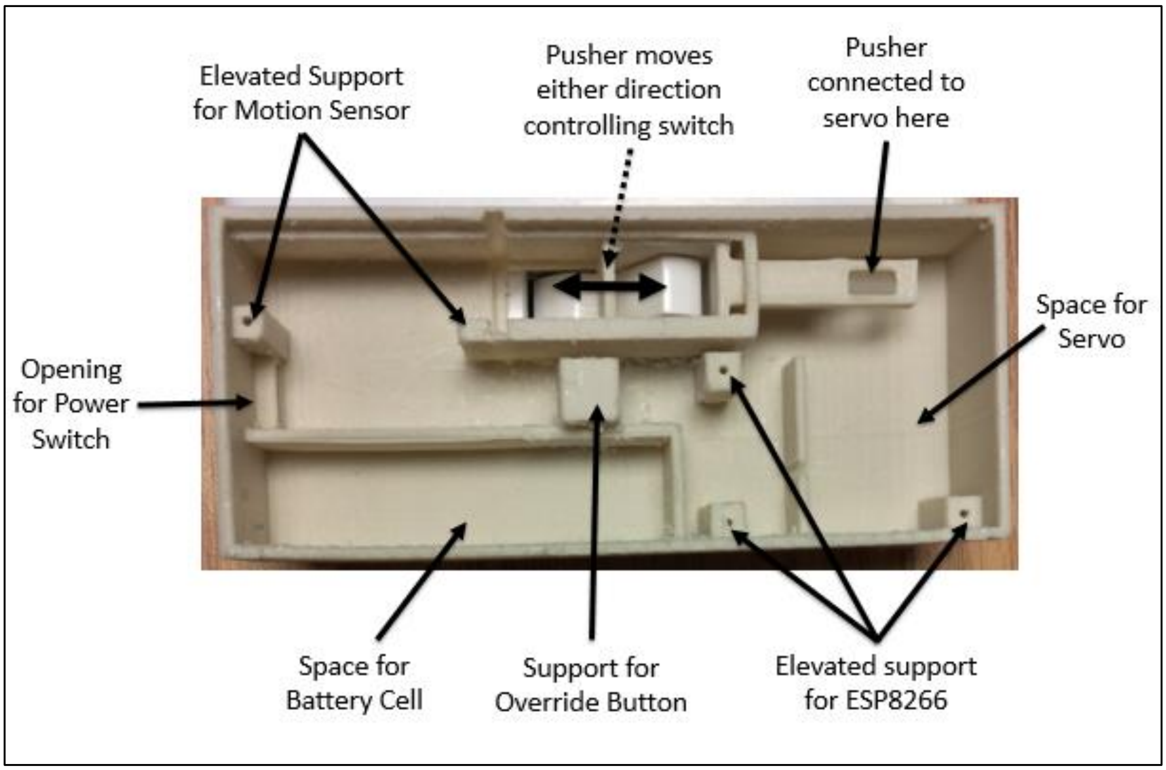


Figure 8: Component placement inside printed 3D body.

The elevated supports inside the device, as shown above in Figure 8, are used to install and position various components inside the device. The elevated supports for the ESP8266 allow the chipboard to stay about 0.5cm above the servo when installed. This leaves enough room for wires to pass through while avoiding any contact with the servo that could potentially damage the embedded system. Similarly, the elevated supports for the motion sensor and the push button ensure that these components fit exactly such that their ‘face’ protrudes from the opening in the front cover when attached.

II. Internal Circuit

Embedded System

For the embedded system in the device, ESP8266 was used which has built-in Wi-Fi connectivity. The ESP8266 is soldered on to a Huzzah Board[3] which equips the ESP8266 with a voltage regulator and a reset button. A 3.3V regulator is needed for proper functionality since the ESP8266 is very sensitive to voltages above 3.3V and can easily malfunction. Figure 9 below shows the ESP8266 soldered on to a Huzzah board. ESP8266 can be programmed using Arduino language in Arduino IDE.

Figure 9: ESP8266 soldered on to Huzzah board.

ESP8266 has 7 digital (PWM) I/O pins on it and one analog I/O pin. Of the 7 digital pins, 1 is used for the servo, 1 for the LED, 1 for the motion sensor and 1 for the override button. Hence 4 digital pins are used by the circuit components.

Motion Sensor

A passive-infrared motion sensor is used in order to detect motion around the switch. This motion sensor is shown below in Figure 10. The motion sensor has two potentiometers on it which are used to control its sensitivity and the timer on the sensors. The timer on the sensors controls for how long the output of the sensor reaming HIGH once motion is detected by the sensor. If the potentiometer is turned all the way to the end (clock-wise), the timer pulls the output HIGH for 50 minutes after motion is detected. If the timer is turned all the way to the other end (counter-clockwise), the output stays HIGH for 2.5 seconds

[3] Sold by Adafruit

from when motion is detected. For this device, the timer was set to a minimum value of 2.5 seconds and further timer settings were applied via coding.

Figure 10: PIR Motion Sensor

The motion sensor can be powered by 3.3-5V, however, it was noticed that the results of the sensor at 3.3V were not satisfactory. Hence, 3.7V was supplied to the sensor directly from the 3.7V battery. The final schematic for the internal circuit of the device is shown below in Figure 11:

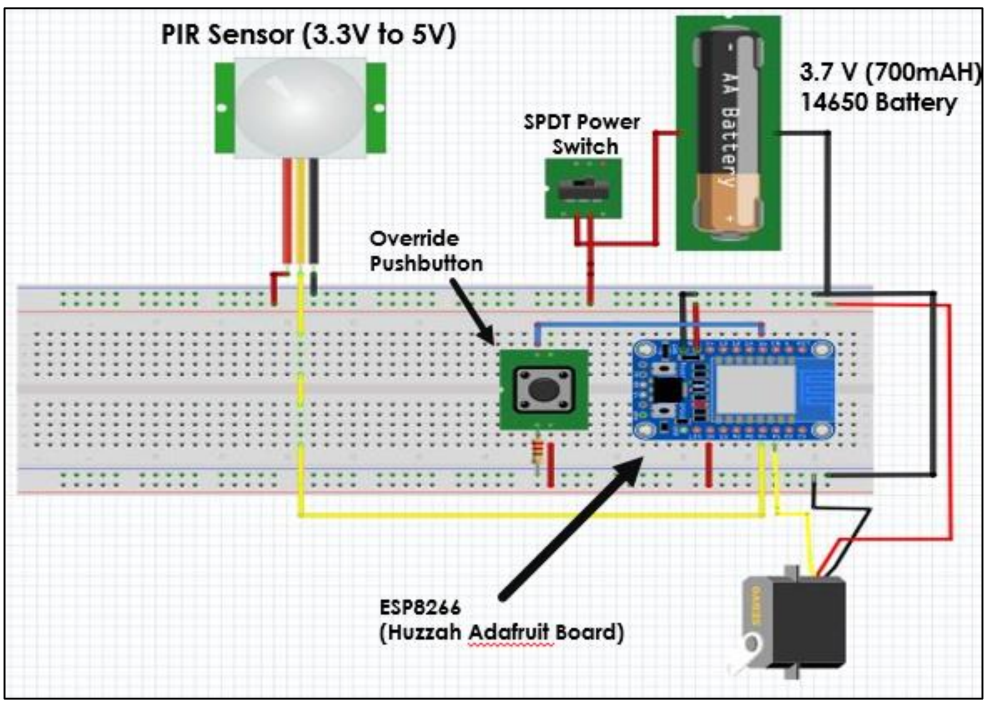


Figure 11: Schematic for internal circuit.

The SPDT power switch controls the battery connected to the device. This switch can be used to power the device ON or OFF. The override button overrides any incoming requests from web server and can be used to control the switch without having to use the web server. Both, the motion sensor and the servo, were powered directly from the 3.7V battery since they were not functioning properly at 3.3V. The servo was unable to exert enough torque at 3.3V and hence it was connected to 3.7V at which it worked as expected. Figure 12 below, shows the device components installed inside the 3D printed device body:

Figure 12: Components installed inside the 3D-printed body.

III. Arduino Code (ESP8266)

ESP8266 can be programmed via Arduino IDE. Arduino's Wi-Fi, Servo and JSON libraries were used in order to implement certain functions. The Wi-Fi library allowed connection to the internet so that HTTP requests could be sent to the server. The servo library was used to control the servo while the JSON library was used to decode the data sent from the web-server. Figure 13, on next page, shows an overview of the Arduino loop that was run on the ESP8266. When powered ON, the ESP8266 first creates an ID for itself based on its MAC address (unique for every device). This ID is sent every time the device tries to connect to the web server. After the ID has been created, it tries to connect to Wi-Fi using the Wi-Fi credentials stored on its memory. If it fails to connect to the stored Wi-Fi, it enters into access point mode where it transmits its own Wi-Fi. The user is then expected to connect to this access point. Once connected, the user will be able to view all the Wi-Fi networks in ESP8266's range. The user will be asked to select one of the available connections and enter its password. The device will then attempt to connect to that connection. After the device is connected to a Wi-Fi network, it performs three steps over and over again:

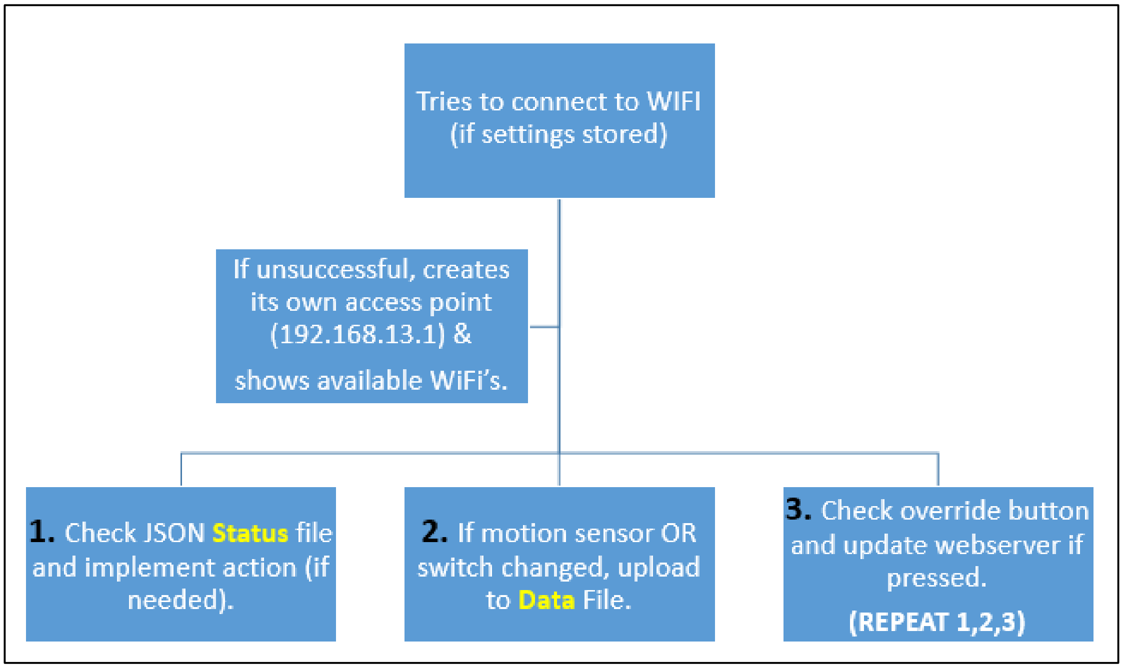


Figure 13: Block diagram showing main Arduino Loop.

1. It connects to the webserver and opens up the **Status JSON file**. Data from the JSON file is extracted and decoded. If the requested status in the JSON data is not the same as the present status of the switch/device, it acts on the request and turns the switch ON or OFF.
2. Next, the ESP8266 reads from the motion sensor if any activity was recorded. If activity was recorded at the motion sensor, or the switch was actuated, the ESP8266 uploads data to the **Data File** on the web server.
3. At the end of every loop, the device check whether the override pushbutton was pressed at any time during the loop. If it was pressed, the device acts and changes the device's state to the opposite of what it is at that moment. The device, then, uploads this activity to the web server. **(Keeps repeating Steps 1,2,3 in a loop)**

IV. Server/Web Page Design

In order to control and access the robotic switch device remotely, a web server is set up using a web hosting provider[4]. This web server uses PHP scripts to accept HTTP GET requests from the device and saves the data in JSON format. An HTML page is used to show device and motion sensor's recent activity from a web browser. The HTML page also allows the user to control the switch. A high-level overview of the web server design is shown below in Figure 14:

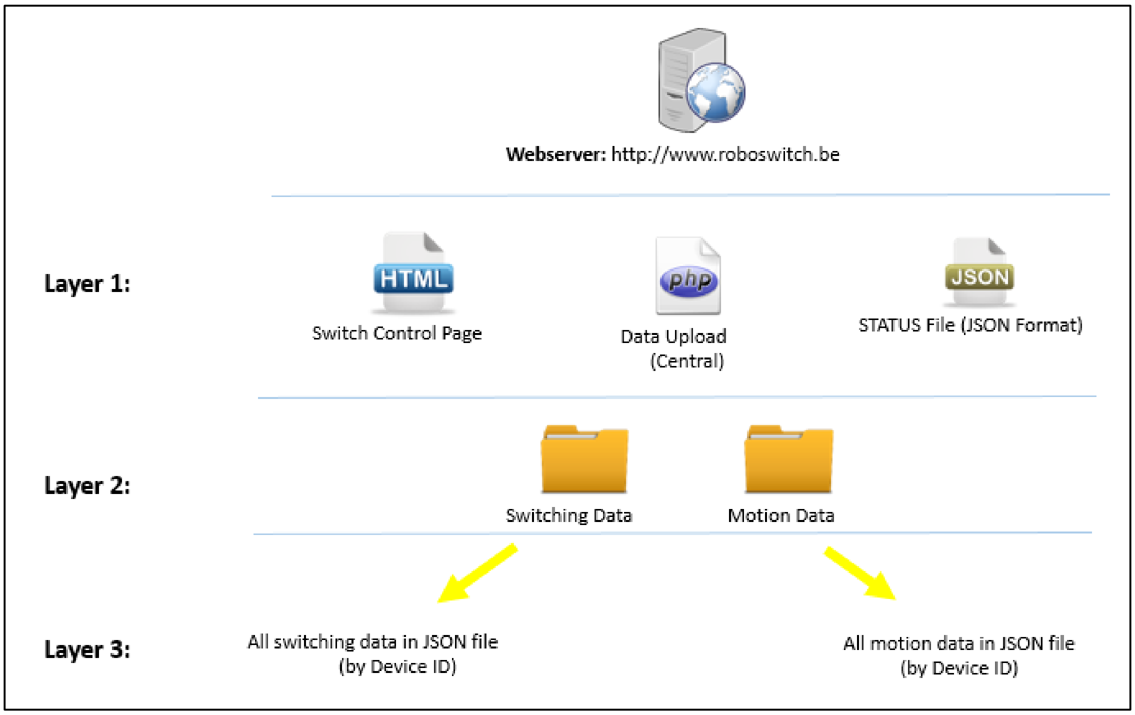


Figure 14: High-level Overview of Web Server Design

The web server design is broken down into 3 layers. The first layer is the only layer that is accessible from outside the server. Any of the files in Layer 1 can be accessed using their URL address. The files inside layers 2 and 3 can only be accessed from within the web server by any of the files in layer 1.

[4] Web Hosting Provider: www.hosting24.com

Layer 1 contains an HTML file, a PHP file and a JSON file. The PHP file is the central file which communicates with the robotic switch device. Any time the robotic switch device attempts to communicate with the webserver, it will be redirected to this PHP file. This PHP script is expecting variables such as ***Device ID, Motion*** *and* ***Status*** in the URL header of the HTTP request. An example of an HTTP request sent by the device to indicate that motion has ended along with the duration the motion lasted (Unix time) is shown below in Figure 15:

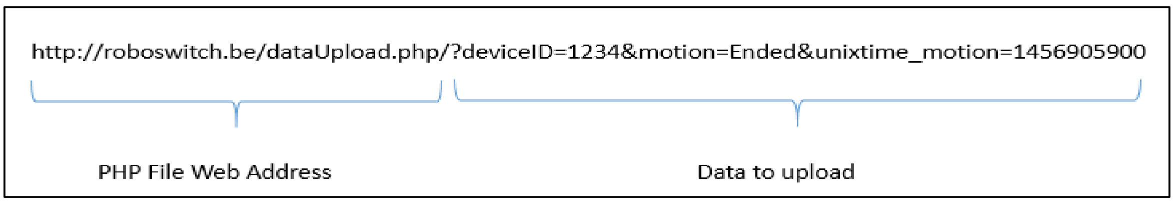
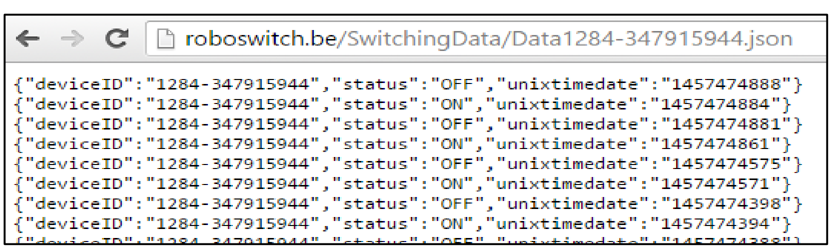


Figure 15: HTTP GET Request (Example)

Each request sent by the device must begin with its device ID since the PHP script sorts out requests using the ID. In the example above, the PHP script would add this event to a JSON file named after the device's ID in the **Motion Data Folder** in layer 3. Similarly, a switching event sent by the device will be added to a JSON file named after the device's ID in the **Switching Data Folder** in layer 3. In case a JSON file corresponding to the device's ID does not exist, the PHP script will create the file automatically. This allows for the addition of an infinite number of devices without any additional coding or modification. Every time the PHP script adds data to a JSON file, it also saves the Unix time the request was received at along with the received data. Figure 16 below shows how the data is stored inside a JSON file containing switching activity:


roboswitch.be/SwitchingData/Data1284-347915944.json

{"deviceID":"1284-347915944","status":"OFF","unixtimedate":"1457474888"}
{"deviceID":"1284-347915944","status":"ON","unixtimedate":"1457474884"}
{"deviceID":"1284-347915944","status":"OFF","unixtimedate":"1457474881"}
{"deviceID":"1284-347915944","status":"ON","unixtimedate":"1457474861"}
{"deviceID":"1284-347915944","status":"OFF","unixtimedate":"1457474575"}
{"deviceID":"1284-347915944","status":"ON","unixtimedate":"1457474571"}
{"deviceID":"1284-347915944","status":"OFF","unixtimedate":"1457474398"}
{"deviceID":"1284-347915944","status":"ON","unixtimedate":"1457474394"}

Figure 16: JSON Data File (Switching Activity)

Similarly, Figure 17 shows how data is stored inside a JSON file containing motion activity. As can be observed, when the event corresponds to motion ending, the duration the motion lasted for is also saved.


```
{"deviceID":"1284-347915944","motion":"Started","timedate_motion":"1457458821"}
{"deviceID":"1284-347915944","motion":"Started","timedate_motion":"1457456639"}
{"deviceID":"1284-347915944","motion":"Started","timedate_motion":"1457161837"}
{"deviceID":"1284-347915944","motion":"Ended","timedate_motion":"1457161438","last_duration":"1463"}
```


Figure 17: JSON Data File (Motion Data)

The **Switch Control HTML File** in layer 1 is the HTML page that allows a user to access all the features the device is equipped with. This HTML page shows most recent switching and motion sensor activity and also lets users control the connected device. Each HTML page is affiliated with a device ID corresponding to the device it can control. This page also shows the current status of the device and shows the last time the device synced with the web server. Figures 18-19 shows two section of the main web page:

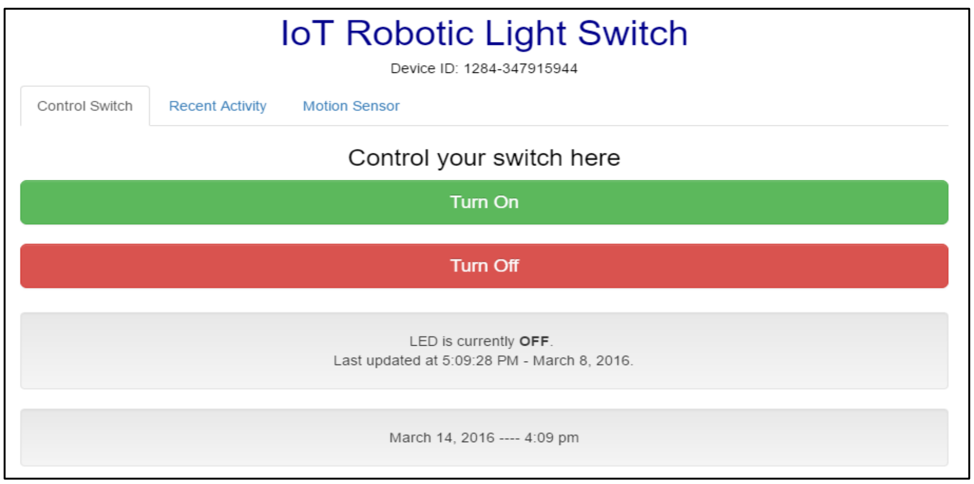


Figure 18: Web Page (Device Control Page)

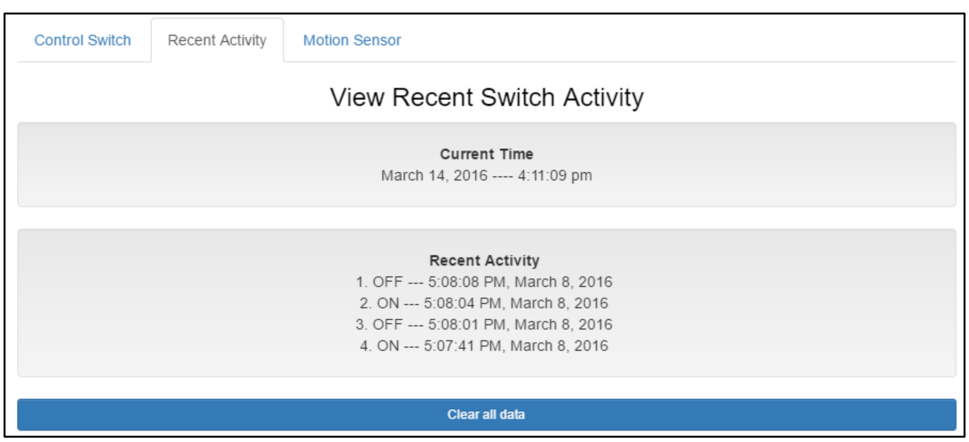


Figure 19: Web Page (Recent Activity)

V. Integration with 3rd Party Services (IFTTT, Cortana, etc.)

The web server can be used to integrate the robotic switch device with other IoT platforms and services. One example is IFTTT[5] platform, with which the web server could be connected using the HTTP URL address of the switching ON and OFF action. Hence because of this integration, the device could be triggered ON or OFF based on hundreds of triggers available on IFTTT website. One example is setting up a trigger that uses the user's smartphone GPS. Based on the chosen settings, the platform could trigger the switch ON or OFF every time the user's GPS suggested the user was within a few hundred meters of the room in which the switch is attached. This would allow the user to implement proximity-based switching.

Another way the device can be integrated with other services is by integrating it with Microsoft's digital voice assistant Cortana. Cortana can be programmed to call HTTP GET requests when a pre-set voice command is given to the assistant. This was implemented with the robotic switch and Cortana was set to call an HTTP GET request based on the user's speech. As a result, when Cortana detected speech 'Trigger, turn the switch ON', it sent an HTTP GET request to turn the switch ON and so the device turned the switch ON. Similarly, this was also done with turning the device OFF.

The above two integrations are just two examples of the possibilities that lie ahead. Based on the open-source design of the robotic switch, it can be integrated with any other IoT platform and programmed to work according to the needs of the user. The motion data recorded by the device can also be used by other devices or services in the room to increase efficiency or implement smart features.

[5] IFTTT (if-this-then-that) is an IoT platform that allows users to automate events based on set triggers.

Section 6: Performance Estimates and Results

Performance

The IoT Robotic Switch functions as expected, however, there were differences observed in the efficiency of the performance of the switch when controlled through different mediums such as the override button, the web server and the voice assistant. In order to test the efficiency, the time between when the request to change status of switch was submitted and when it was completed was recorded. It was observed that it took longest when request was made via digital voice assistant while it took the shortest amount of time when the request was submitted through the override button. However, this is expected since the override button does not involve submitting any requests online and hence the switching is done instantly. Table 3 below shows the average time taken for the request to be fulfilled via various mediums:

Table 4: Time taken for switching command to be executed

MEDIUM	TIME TAKEN	AVERAGING SAMPLE SIZE
OVERRIDE BUTTON	1-2 seconds	25 attempts
WEB SERVER	3-4 seconds	40 attempts
DIGITAL VOICE ASSISTANT	7-9 seconds	20 attempts

In terms of the non-functional requirements, the device was able to fulfill its objectives. The final web server design was robust and scalable such that multiple devices could be added to the system without having to modify the code. The installation of the device was also extremely simple and needed no wiring.

The device uses 2500mAh LiPo cell (18650 Battery) that provides 3.7-4.0V. The time the device would last on a single battery depends on the usage of the device and on the frequency at which the device syncs with the web server. At maximum frequency (check with web server every 1 second), the device's current consumption ranges from 50 µA to 150mA. This averages to around 5 mA. Based on this current consumption without utilizing sleep-mode features, the device would last about a few weeks.

Suggestions for Improvement

Based on the performance results and estimated noted on the last page, certain improvements must be made to the device to make it more useful from a consumer perspective. First, the device's sleep-mode features should be tested and incorporated into the algorithm. This will dramatically increase the battery life of the system from a few weeks to a few months. At the same time, on-board LEDs which serve no purpose for the device will need to be removed since they can help increase battery life by about 20-30%.

Another way to reduce battery life could be to change the design of the device. This would involve including a connectivity hub that will be directly connected to an electric outlet somewhere near the robotic switch device. The switch device could then connect to the hub via Bluetooth and upload all information to the hub. The hub would then be able to connect to the internet and communicate with the web server. Since the hub would be connected to an outlet, battery-life would no more be an issue. And since Bluetooth consumes drastically less energy than Wi-Fi, the device's battery life would increase as well.

Also, the mechanical design of the system could be further improved. The current design produced a loud noise when the switch was actuated from one position to another. Although this was taken care of to an extent by applying a gel layer inside the device on the surface which the light switch button was striking against, the design could be improved to get rid of this noise completely without having to apply a separate layer.

Section 7: User Manual

Installation:

1. Peel off two 3M Adhesive strips and place them on the back of the IoT Robotic Switch.
2. Take the cover off the device and place it on the light switch such that the toggle button fits into the opening in the 'pusher' component of the device as shown in Figure 8 earlier.
3. Once positioned properly, gently press the device to ensure it sticks on to the switch.
4. Now place the device cover back on to the device.

Using the device:

1. Turn the device ON using the small power switch on the top of the device.
2. Once the LED starts blinking, bring out your smartphone or your laptop and see the available Wi-Fi connections. Connect to the Wi-Fi starting with 'ESP8266 -XXX'.
3. Open the web browser and go to web address, http://192.168.13.1. This page will bring up a list of Wi-Fi connections that the robotic switch can detect around it. Select the network you want to connect to and enter its password. Click 'Connect' and wait for the device......
4. Upon successful connection, the LED will blink thrice and then turn OFF.
5. Now go to the webpage linked with the device: http://www.robotswitch.be/[DeviceID]
6. The page will display the time the device last synced – confirming connection with it.
7. The device can now be controlled via this webpage.

Integrating device with other platforms/services (Cortana, IFTTT, etc.):

- Replace the device ID in the square brackets of the following URL and use it to connect with services such as IFTTT or Cortana: http://www.roboswitch.be/**[#DeviceID]**?status=ON

Section 8: Conclusion

The goal of the project was to design and develop a prototype for an Internet of Things Robotics Light Switch that could be easily installed on top of an existing light switch and would equip the switch with smart features through a web application. A complete working prototype was developed and its features were thoroughly tested to get performance estimates. The functional and non-functional requirements such as easy installation, cost-effectiveness, motion sensor control, integration with other platforms and digital voice assistant control were met.

For future work, the device's battery consumption could be lowered through the use of the embedded system's sleep-mode features or even adding a hub device that could be set up separately at any location in the house. Then the hub device would serve as the medium through which the web server and the device would interact. This would allow the use of an energy-efficient communication mode, such as Bluetooth technology, between the device and the hub in order to increase the battery life.

Similarly, a smartphone application could be developed for the device. This could also be used to enhance the features of the device by introducing analytics based on the device usage data. Also, the device's mechanical design could be altered to create version of the device that could support other styles of switches in addition to the rocker-style switches.

In conclusion, the prototype was successful in achieving most of the desired performance requirements. The developed prototype can serve as a minimum-viable product (MVP) that can be commercialized as a consumer product. The MVP can then be used to gather validated learning which can further be used to develop the design and features of the device.

References I

[1] “Home Automation Market – Global Industry Analysis, Size, Share, Growth, Trends and Forecast 2014 - 2020: Research Report,” by *Transparency Market Research* in Semiconductor & Electronics, pp. 123.

[2] US Energy Information Administration, http://www.eia.gov/tools/faqs/faq.cfm?id=99&t=3, 2010.

[3] US EIA – Utility, http://www.eia.gov/electricity/sales_revenue_price/pdf/table5_a.pdf, 2014.

[4] Gregg, Matt, “How Many Light Bulbs Are in The Average American Household?” in *Home Improvement*, 2011.

[5] US Department of Energy – Higher Education, www4.eere.energy.gov/, 2015.

[6] Thota, K., “Broadband and Wi-Fi Households Global Forecast 2012,” *Strategy Analytics*, 2012.

[7] Smith, Aaron. "US Smartphone Use in 2015.” by *Pew Research Center*, 2015.

[8] comScore. "Subscriber Share Held by Smartphone Operating Systems in The United States from January 2012 to September 2015." *Statista - The Statistics Portal*, Statista, 2015.

[9] HobbyKing – HXT900, http://hobbyking.com/hobbycity/store/uh_viewItem.asp?idProduct=662.

[10] *Espressif Systems,* ESP8266, 2013 pp. 10.

[11] van Diggelen, F., "Indoor GPS theory & implementation," in *Position Location and Navigation Symposium, 2002 IEEE* , vol., no., pp.240-247, 2002.

Appendix A: Business Model Canvas

I. What is a Business Model Canvas?

The Business Model Canvas (Blank, 2012) is a tool used to summarize the key aspects of an idea or a startup. The canvas is based on the idea of a lean startup (Ries, 2011). Lean startup is a relatively new phenomenon for developing businesses and products that was first introduced in 2008 by Eric Ries. This method is based on the claim that startups can shorten and improve their product development cycles by business-hypothesis-driven experimentation, iterative product releases and an idea validation approach. In a nutshell, this approach calls for startups to begin with a minimum viable product that is only equipped with the most basic features what would be traditionally termed an initial prototype. Based on the user feedback and demand, the lean startup method claims, startups should perform iterative product releases focusing on adding and improving on the features that are actually wanted by their customers. This approach validates the idea the startup is based on during the initial stages and helps entrepreneurs align their activities to best reflect the customer demand. At the same time, this approach also saves entrepreneurs and technologists from needing large amounts of initial project funding reducing the capital at risk.

The Business Model Canvas consists of components such as value propositions, customer segments, customer relationships, channels, key activities, key resources, key partners and the revenue streams and cost structure. Hence, this canvas provides a basic idea and a summary of the entire idea of the startup. Below is a description of each of the component in the canvas:

Value Propositions:

This refers to the key proposition that the startup or the idea offers. This would not only be the core idea the startup is based upon but also the distinguishing feature of the startup over any existing companies.

Customer Segments:

This sections mentions the target customer segments of the idea or the startup. The segment could either be broken up into socio-economic segments or by profession, age, demographics, ethnicity or any other characteristic that helps, to an extent, generalize the behavior of the segment. This helps understand and decide the requirements of the core product so that it is catered according to the demand of that segment. Hence, it becomes extremely critical to understand the behavior of the intended customer segment or segments. *Customer segment* and *value proposition* parts of the canvas form the core of any business canvas and often are the first two components filled out. These two sections define the whole concept of the startup.

Key Activities:

This component of the canvas lists down the activities that are at the core of the business or the startup. This section is often looked at in conjunction with the *value proposition* component in order to ensure that the firm is only focusing on the activities that directly influence its *value proposition*.

Key Resources and Key Partners:

Both these components of the canvas allow the business to consider the key partners and the resources that are enabling it to perform the *key activities* in order to deliver its *value proposition*.

Customer Relationships and Channels:

These components of the canvas allow the startup to further understand how it is reaching out to its intended customer segments. *Customer relationships* section also contains the activities of the firm that are enabling it to maintain its customers and ensure that the relationship with the customer is properly handled. The *channels* section allows the startup to understand how its *value proposition* is delivered to its various *customer segments* and hence serves as the bridge between the two.

Revenue Streams:

After having dealt with all the features above, the canvas moves on to the monetary aspects of the business. This is where the startup figures out whether it can deliver the value proposition and perform the activities through its partners at a cost that is **lower** than the revenue it generates. The revenue stream section contains information on how the business makes money. This section investigates various customer revenue models such as pay-per-use or pay-per-product or a mix of the two. While deciding the revenue stream structure, a business must keep in mind the preference of its customer segments.

Cost Structure:

Lastly, the canvas considers the costs of the startup. This section takes into account the *key partners, the key activities* and costs associated with maintaining *channels* and *key resources*. As a result, an investor or an entrepreneur can find out if the idea is sustainable or not. It is interesting to note that the monetary aspects of the business are only considered at the end of the canvas. This is because in the lean startup method, the initial cost is extremely low and the other components of the canvas are really being explored. The lean method considers the early stage in the life of a startup as the period where the startup is filling in these boxes and finding the right answers. Once the startup has filled in these boxes, it is really in a position to see whether the idea is sustainable or not!

II. Business Model Canvas for an IoT Robotic Light Switch

In this section, a Business Model Canvas is developed for the proposed IoT Robotic Light Switch. This model will help further develop the idea and help validate its viability as a potential profitable business idea.

i. <u>Value Proposition</u>

The IoT Robotic Light switch provides customers an option to transition to a smart home (lighting) through a **phased implementation approach** that simplifies the installation process by **getting rid of any wiring**. At the same time, the Robotic Light switch equips the toggle light switch with all capabilities of a smart switch. The light switch also allows **integration** with other smart home devices.

Phased implementation refers to the ability to install the light switch only at points where the user feels a need to. Hence, the user would be able to transition slowly to an increasingly automated home. This is especially great for people who are not sure of the benefits of smart home lighting systems and just want to try it without huge up-front costs.

ii. <u>Customer Segments</u>

This device would appeal to mainly three customer segments:

- <u>People living in apartments:</u>

  Such customers usually pay their own bills and realize the cost savings that could be achieved by switching to a smart lighting solution. However, since they do not own the apartment or the room, it does not make sense for them to install a typical smart lighting solution that tampers with the wiring. Additionally, they might not be allowed to make such upgrades as a part of their leases. This device would present them with a solution that would not tamper with the existing switches and at the same time offer them a smart lighting solution. Also, they would be able to keep the device with them when leaving the apartment and moving to a new place.

- Homeowners on a budget:

  This solution is not for people who have unlimited funds at disposal to entirely revamp their house and install a more traditional smart lighting solution. However, homeowners who are on a budget and who care about the economic outcomes of their decisions would be a target customer for this light switch. This device would allow them to automate the most frequently used lights in their house and save them cash since they won't have to make a huge investment revamping the entire lighting system in their house. This would also give them a chance to see whether home automation is a viable solution for their problems since it will leave the option of reverting back to the old system open.

- Technology enthusiasts and DIY-ers

  This device would also appeal to DIY-ers who are thrilled about the idea of a device that is open-sourced and can be integrated with any other smart-house device.

iii. Key Activities

The key activity the startup would be involved in is managing the device production stage. This will involve working with partners to receive all the device components and then finally assembling the switch device. The web server that supports the device will also have to be managed. Another key activity would be R&D especially with a focus on integrating the product with other home automation products and platforms. Finally, regular feedback from device users will be used to create future revisions of the product.

iv. Key Partners

The partners are derived from the key activities mentioned above and also from the channels and customer relationships. The manufacturing and assembling of the device will have to be partially outsourced. The PCB board inside the device will be outsourced to a PCB-printing who will be one of the key partners. Similarly, the servos inside the device and the override button will also be outsourced and bought from a key partner. The device case will be printed via plastic injection molding by a key partner.

For advertisement and selling (marketplace) purposes, Amazon and Facebook will be key partners. The product will be sold on Amazon's website while it will be advertised on Amazon, Facebook and various search engines. A web server will have to be maintained and a company providing web hosting solutions will be another partner. Table 5 below summarizes the key partners according to the functions they provide:

Table 5: Key partners and their main function.

Activity for which partner needed	Partner
PCB Circuit Board	PCB Printing Company (customecircuitboards.com)
Servo, Buttons	Electric Components Producer (seller from Alibaba)
Plastic Injection Molding	Initial Mold Producer (Shanghai Sourcing)
Web Hosting	Web/server hosting services (Amazon Web Services)
Advertisement/Market Place	Amazon Marketplace/Facebook/Google
Mailing / Delivering	USPS & UPS

v. Key Resources

The PCB circuit design and the plastic mold (3D CAD) design are two of the most important resources. Similarly, the code that will be loaded on the devices and the web hosting code that is used to run the servers is a very important key resource. Lastly, the tools required to assemble the device after receiving parts from various partners will be another key resources.

vi. Customer Relationships

The idea behind the IoT Robotic Light switch is that it would give users an opportunity to implement a phased transition to a smart house. Hence, the users would test a sample of the robotic switch before buying more such switches. For this business model to be successful, it will be extremely important to ensure a smooth customer relationship which will hopefully allow the company to grow its customer-base. In order to achieve this, a 24/7 phone line would be set up to assist customers and special attention would be given when preparing product manuals and instruction packets. Since the device will be battery powered, installing/replacing batteries could be a potential barrier for customers when deciding to purchase these devices. Hence, a system will be set up to provide discounted batteries to customers through the first 3-4 years of the device's life. In order to achieve this, the device will also have capabilities that would allow the startup to keep track of remaining battery life of the device.

vii. Channels

Channels will be extremely important in order to deliver the value proposition to the targeted customer segments. This will involve reaching out to and conveying the device's value proposition to potential customer segments. In order to accomplish that, Amazon's market-place will be a key channel. Also, Facebook's and Google's advertisement services will be essential in order to advertise the product. Another channel would be the existing customers themselves. They could be given some sort of incentives if someone buys the device through their referral. There will also be a company website that directly sells these switches.

viii. Revenue Streams

The home automation industry has not yet seen too many players that offer pay-per-use revenue structure. And since the proposed robotic switch has a major cost advantage to existing solutions, adopting such a revenue structure could complicate things and miss out on this advantage. Therefore, adopting a pay-per-use model would complicate the decision making process for customers, which would mean missing out on an opportunity to clearly distinguish the cost-effectiveness of such a product as compared to alternatives.

Thus, the revenue model for this product would be a pay-per-product for the most part. However, there would be an extra optional feature providing the users with analytics of the data generated from the device that would be charged at a monthly price. The users will still have access to all their data (free of cost) if they wanted to use it with another platform or service. After taking into account the cost analysis and various customer surveys for this device, the price point that has been determined for this device is around the $27 mark. A further breakdown of the cost of the device is given in the cost structure section later. The additional pay-per-use analytics feature would be priced at around $3 per month and some discounts would be made available for using the feature with multiple devices.

ix. <u>Cost Structure</u>

For the initial phase of this startup, the costs would mainly be restricted to the cost of producing the good, cost of the webserver, cost of advertisement and the cost of shipping the product. However, in the future, costs such as employing programmers for either smartphone app development or further website development will have to be considered. Similarly, the costs to rent a storage space and a development space would also have to be accounted for in the future. For the initial stages, labor costs are also not considered since it will be mostly provided by the owner or key partners. Table 6 below breaks down various costs associated with production of the device. These costs are per unit and also depend on the quantity of parts ordered since buying in bulk reduces the cost. (Mold is only a one-time cost while mold injection is a recurring cost per unit)

Table 6: Cost of components based on quantity ordered.

	Device Components ($)					**Device Case ($)**	
Quantity	**Servo**	**Buttons**	**PCB Print**	**Motion Sensor**	**Battery**	**Mold Injection**	**Mold (One Time)**
30-50	1.50	0.40	3.00	1.50	3.00	3.00	3,000.00
50-300	1.40	0.40	2.50	1.25	3.00	3.00	3000.00
400+	0.90	0.30	2.00	1.00	3.00	3.00	3000.00[6]

Using these component costs, the average variable, fixed and total costs were derived as shown below in Figure 20. Cost of setting up the mold is by far the biggest cost. However,

[6] Component costs confirmed from Alibaba.com vendors.

since it is a one-time only cost, as production scales up, its effect on the overall average cost decreases. It can be observed that as the quantity of units produced increases, the average unit cost approaches to around $8.26 as the one-time cost of the mold is offset. Using this analysis, for pricing purposes, the cost of each unit can be estimated to be around $10.

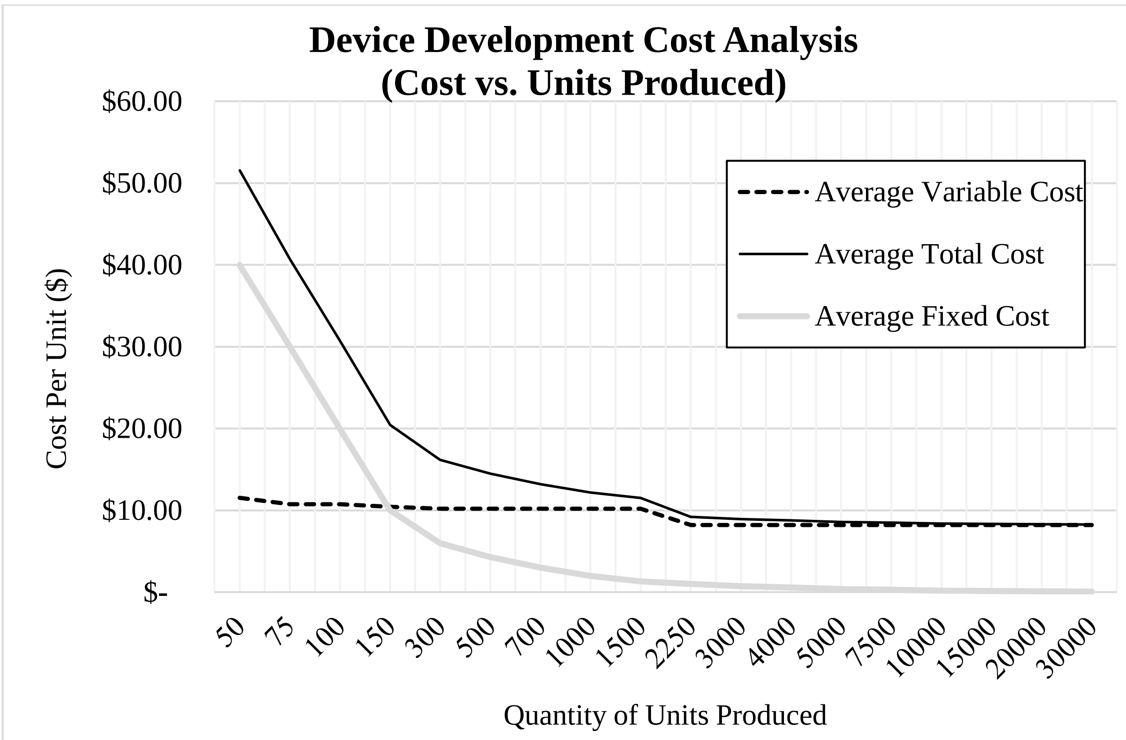


Figure 20: Device production cost analysis vs. quantity produced

Besides the cost of product development, the other costs that need to be considered are web hosting costs, advertisement costs and shipping costs. For web hosting, there are two options: shared web-hosting and virtual private server (VPS). For the initial phase of the company life-cycle, shared web-hosting should be sufficient. Usually, VPS is only needed when there is ultra-high traffic on the website. Shared web-hosting can cost around $50 a month. Another $100 will be appropriated for advertising costs annually. Shipping will cost about $5 per device shipped through USPS. These costs are added to the cost analysis and Figure 21 is generated to reflect cost of shipping, advertising and web hosting in addition to the device development cost in Figure 20. This is called the **cost of selling one unit.**

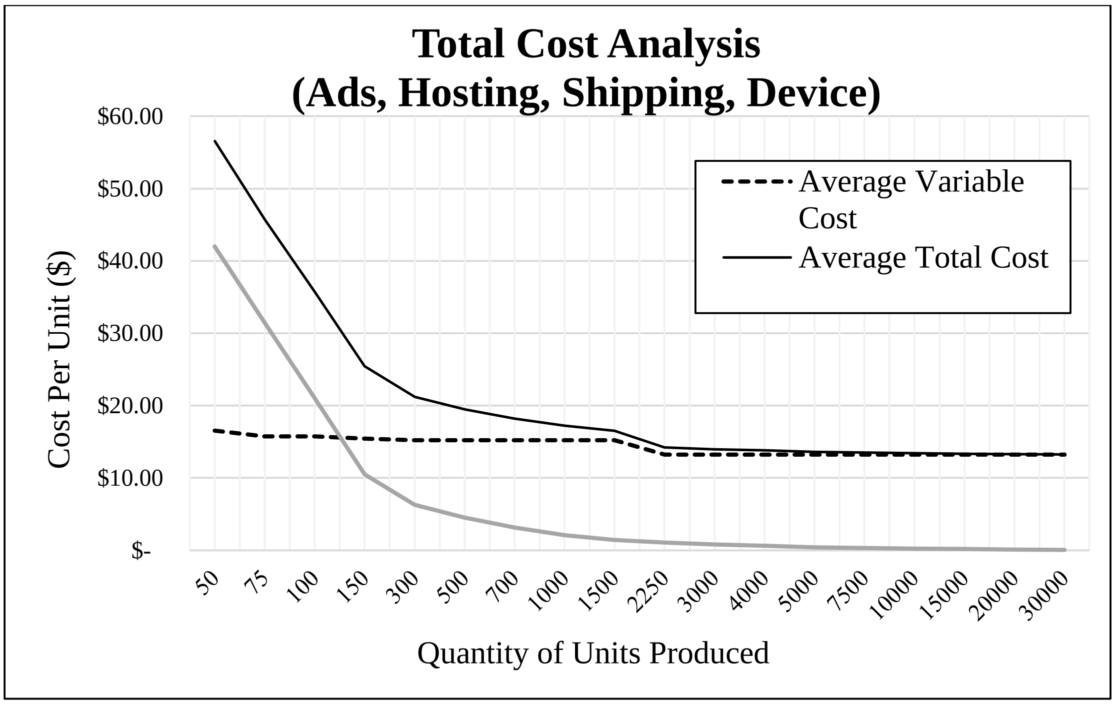


Figure 21: Cost analysis considering shipping, web hosting, advertising and device costs.

Based on the analysis, it costs around $18 per unit when the volume is around 1000 units and $13 per unit when the volume is around 10000 units. Table 7 below represents the estimated **cost of selling** one unit at various quantities:

Table 7: Average cost of selling 1 unit given various quantities.

Volume	Average Cost of Selling 1 Unit
100	$ 45.75
300	$ 25.45
500	$ 21.20
1000	$ 18.20
3000	$ 14.20
10000	$ 13.50
50000	$ 13.26

Based on the cost-analysis above, a **price point of $27** would cause the company to break-even as far as the total quantity of units sold is above 150. When the quantity of units sold is around 300, the profit margin is at 6%. However, at around 1000 units the profit margin goes to around 50%. A price of $27 also compares well with the price of the closest competitor of such a robotic switch. WeMo Light switch is a similar smart switch, however, it has to be wired and installed by a professional technician. The WeMo switch costs $49[7] and hence the $27 price clearly distinguishes the robotic light switch in terms of cost-savings.

[7] Price quote obtained from Belkin's website.

References II

- Bauer, Harald, Mark Patel, and Jan Veira. "The Internet of Things: Sizing up the opportunity." *McKinsey at http://www.mckinsey. com/insights/high_tech_telecoms_internet/the_internet_of_things_sizing_up_the_opportunity (visited 2/8/2015)* (2014).
- Blank, Steve. "How to Build a Startup - The Lean LaunchPad." Udacity. Ed. Kathleen Mullaner. Udacity, 23 Aug. 2012. Web. 10 Mar. 2016.
- Brush, A. J., Bongshin Lee, Ratul Mahajan, Sharad Agarwal, Stefan Saroiu, and Colin Dixon. "Home automation in the wild: challenges and opportunities." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2115-2124. ACM, 2011.
- Middleton, P., P. Kjeldsen, and J. Tully. "Forecast: The Internet of Things, Worldwide, 2013, Gartner." *Inc. Nov* (2013).
- Ries, Eric. The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses. Crown Books, 2011.
- Tolentino, Mellisa. "What Quirky’s Bankruptcy Means for the Smart Home Hub Market." Silicon Angle. N.p., 30 Sept. 2013. Web. 09 Mar. 2016.
- "Who's embracing home automation and why: what's driving the adoption of smart home products?" Air Conditioning, Heating & Refrigeration News 18 Jan. 2016: 14+. Business Insights: Essentials. Web. 8 Mar. 2016.