

Final Year Project/ Thesis (2024 – 2025)

ELEC/CPEG **Model** Proposal Report*

Smart Light Switch

Project ID: AB04a-20
Supervisor: Professor. A
Author (Student ID): CHAN Tai Man (12345678), CHAN Siu Ming (23456789)
Date: 17 September 2020

Main Objective

This project aims to design and build a small wifi-connected external smart switch that has high functionality while improving compatibility, and at the same time makes smart lighting more user-friendly and attractive to users. We name it the External Light Switch Assistant (ELSA). It is a device that can be fixed to an existing switch in the home and can be operated using a mobile phone. It allows users to turn the switch on and off remotely and manually, whether inside the home or within wifi range. It provides scheduled and timed lighting and other IoT-related features.

Objective Statements

1. To create the switch mechanism of ELSA that fits on a standard existing electrical switch in Hong Kong and can activate the existing switch to turn it on and off as required. The switch mechanism should activate in under 1 second.
2. To program the microprocessor to link ELSA to a web server so that ELSA can be accurately controlled through a web browser.
3. Set up a configuration system, implement other features, and create a user interface for ELSA so that ELSA can discover new devices and the user can configure WiFi settings and set automated tasks.

*Note that the authors of this report have given permission for it to be modified from its original version to reflect the current final report format, and for some of the content to be changed for illustration purposes.

Contents

SECTION 1—INTRODUCTION	0
1.1 Background and Engineering Problem.....	0
1.2 Objectives.....	0
1.2.1 Objective Statements.....	0
1.3 Literature Review of Existing Solutions.....	1
SECTION 2—METHODOLOGY.....	2
2.1 Overview of ELSA	2
2.1.1 System Description	2
2.1.2 ELSA Diagram	2
2.1.3 Components List.....	3
2.1.4 ECE Knowledge.....	3
2.2 Objective Statement Execution	4
2.2.1 Switching mechanism and external device.....	4
2.2.2 Set up Hardware and Software	6
2.2.3 Set up a configuration system and implement other features.....	8
SECTION 3— Project Planning.....	12
3.1 Project Schedule	12
3.2 Budget.....	14
REFERENCES	15
APPENDICES	16
Appendix A – Meeting Minutes	16

SECTION 1—INTRODUCTION

1.1 Background and Engineering Problem

The Internet of things (IoT) is the network between devices and systems; they can exchange data with each other through the internet. It is widely used in different aspects and home automation is one of the examples. It refers to remote and automatic control of electronic appliances via mobile devices. These devices should be programmable and can access a communication network [1]. Smart homes have become increasingly popular because they provide domestic convenience and safety. A few clicks on a smartphone in the user's hand can switch off all home appliances, saving them time to switch them off one by one in different rooms. Even if people forget to turn off an appliance before leaving the home, they can still turn it off outdoors, which not only saves the time needed to return home, saves a lot of wasted energy, and prevents accidents caused by over-operated devices such as irons and heaters. Combining home assistant devices, customized scheduling and sensors improve the functionality of home appliances.

Commented [1]: Introduction into the broader area of research.

The most common type of home automation is smart lighting. Currently, there are several smart lighting products available on the market. However, these products are quite expensive, usually costing from \$20 to \$70 us dollars each depending on the functionality. Also, they have limitations in different aspects. Some of them have complicated installation procedures, users have to remove the existing light switch and do wiring connections, which can be dangerous, or users need to pay extra money to find corresponding technicians for help to install. Some of them lack compatibility so they cannot be used with existing lighting hardware. Therefore, they may not be attractive to people who have never used them but want to try smart lighting. This discourages people from utilizing this useful technology. There is a need to develop a smart lighting device that is cheap, easy to install without changing the existing home systems, and fits most lighting systems.

Commented [2]: More focused area of research.

Designing a smart home device to control lighting that satisfies these requirements is vital. A cheap and basic device that attaches to existing switches and has simple installation procedure that ensures people do not need any knowledge about electronics can solve these problems. This can help more non-technical persons with a small budget to have access to home automation or IoT. A smart lighting device that is cheap, easy to install without changing the existing home systems, and fits most lighting systems will ensure smart lighting is more ubiquitous and that will help save energy and the environment.

Commented [3]: Identification of the problem/s.

Commented [4]: Impact of solving the problem/s

1.2 Objectives

1.2.1 This project aims to design and build a small wifi-connected external smart switch that has high functionality while improving compatibility, and at the same time makes smart lighting more user-friendly and attractive to users. We name it the External Light Switch Assistant (ELSA). It is a device that can be fixed to an existing switch in the home and can be operated using a mobile phone. It allows users to turn the switch on and off remotely and manually, whether inside the home or within wifi range. It provides scheduled and timed lighting and other IoT-related features. **Objective Statements**

1. To create the switch mechanism of ELSA that fits on a standard existing electrical switch in Hong Kong and can activate the existing switch to turn it on and off as required. The switch mechanism should activate in under 1 second.
2. To program the microprocessor to link ELSA to a web server so that ELSA can be accurately controlled through a web browser.
3. Set up a configuration system, implement other features, and create a user interface for ELSA so that ELSA can discover new devices and the user can configure WiFi settings and set automated tasks.

Commented [5]: Note that this objective is measurable. Testing can be done to obtain results and the result of the test can be compared to the desired result (measurable objective) to evaluate the success of this objective.

Commented [6]: This is a measurable objective.

Commented [7]: This is a measurable objective.

1.3 Literature Review of Existing Solutions

There are three common types of smart lighting products currently available on the market: Smart Light Switches, Smart Light Hubs, and Smart Bulbs. These devices will be reviewed in the following.

Smart Switches

While keeping the physical switch method to control lights, smart switches additionally provide remote and automatic control. Same as traditional light switches, smart switches are wall-mounted switches. When the physical switch is clicked, it will send a signal to the microcontroller inside. The microcontroller then connects or breaks the circuit between the light and the power supply to turn the light on or off [2]. With connection and accessibility using wireless communication networks such as Wi-Fi, ZigBee, and Bluetooth, users can perform remote control by using the app on a smartphone or tablet, speaking to a smart home assistant like Google Home and Apple Home Kit, or using the remote controller provided. Smart switches can also do scheduled and timed lighting with instructions set in the app.

Smart switches provide flexibility to users to choose suitable operation methods according to their needs. Users who are not able to use their phones or unfamiliar with using smartphones, especially the elderly, can still control the lights physically. However, installation of smart switches is inconvenient: screwing and wiring work is necessary since they are in walls and hardwired. A skilled electrician is needed to install the system, which causes additional installation costs. Moreover, most smart switches require neutral wire connections to give power supply to the microcontroller [3]. Therefore, houses that do not have a neutral wire network, such as old houses in China and the United States have limited choices.

Smart Plugs

Smart plugs are another alternative to achieve smart lighting. It has a working principle similar to smart switches. It contains a microcontroller to control the power flow between the power supply and the appliance, as well as a radio module to connect and access wireless networks [2]. Likewise, users can operate smart plugs by a smartphone app or digital home assistants. Smart plugs have the same functionality as smart switches.

Compared with smart switches, smart plugs are more convenient to install and have greater flexibility. Smart plugs do not require complicated wiring work, they can be installed by simply plugging in or out from a socket. Because of this, it is convenient to change the entire lighting setup. Users can move the plugs' position or convert a different lighting device whenever they want. However, smart plugs are only suitable for plug-in devices and cannot control hardwired ceiling lights, which are usually the major light source of houses. Also, smart light plugs are often bulky in size and may block adjacent sockets, decreasing the number of available sockets for users.

Smart Bulbs

Smart bulbs consist of a bunch of LEDs, a microprocessor to control LEDs, chips, and modules to perform wireless communications [5]. They can be operated through smartphone apps and smart home assistants to control the bulbs. For other functions such as scheduled lighting automation alone, an extra hub is necessary [4]. The advanced version of smart bulbs can even adjust the color and dimness of the light. However, the more functions smart bulbs have, the more expensive they are. Another thing that requires attention is that the light switch needs to be on at all times, or else the smart bulb has no power to operate.

Installing and removing smart bulbs is as easy as smart plugs. Apart from flexible setup, smart bulbs can be used for both hardwired and plug-in lights, solving the weakness of smart plugs. However, if the original light switch controls multiple light bulbs, using a smart switch is more cost-effective than using several smart bulbs. Most smart bulbs use a standard screw base for mounting, so not all lights can use smart bulbs [2].

All three smart light products demonstrate decent functionality. While some have solved the problem of installation, they still have room for improvement in terms of compatibility. Meanwhile, some smart door lock products solve the flaws mentioned above, they are "Sesame Lock" [6] and "August Wi-Fi Smart Lock" [7]. They attach to the original lock and turn the lock in a direct mechanical way. They are powered by a battery and have Bluetooth and Wi-Fi connections. There is no need to replace any part of the lock in the door. The advantage of this product is keeping the original mechanism. Users can use it without any technical knowledge. Additionally, if "Sesame Lock" does not work, the user can remove it easily and use the old mechanism as usual.

This project proposes the design of an external smart switch. It utilizes the existing light switches at home but can still offer the same function as other smart lighting devices. It aims to have high functionality while improving compatibility at the same time to make smart lighting more user-friendly and attractive to users.

SECTION 2—METHODOLOGY

1.1 Overview of ELSA

2.1.1 System Description

In this project, we create a switch mechanism that can switch on and off a standard existing electrical switch and can be controlled through wifi using a mobile phone or controlled manually. The External Light Switch Assistant (ELSA) can be adhered to an existing light switch using double-sided adhesive tape, and can function without anyone holding any components. In the best case, it should fit on two or more brands of a switch in Hong Kong, because some switch designs are similar. The switch will be activated in under 1 second.

There is a hardware and a software control system for ELSA. These allow ELSA to connect to the web server for handling user requests, report switch status, and control the mechanical part to complete on/off requests accurately with the aid of sensors.

A simple interface has the function of turning on and off the switch. While the user can change the configuration in the source code, the target users are non-technical users who do not need to learn to edit and compile the code themselves. To make ELSA more user-friendly, it has a user interface, which allows users to easily configure ELSA to perform complex tasks, like task scheduling.

The main working environment of the switch is home or other places with a stable WiFi network. ELSA does not require an authorization system as the “Sesame Lock” [6] reviewed in the literature review, but it needs to be able to communicate with other switches in the LAN. Therefore, WiFi network access and a method to communicate with other switches are needed.

2.1.2 ELSA Diagram

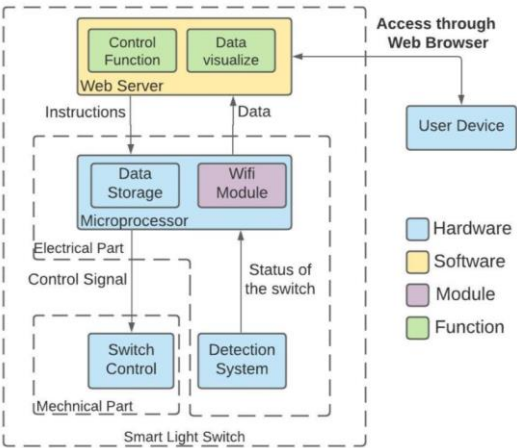


Figure 1. Schematic of the whole ELSA system

Figure 1 shows a basic diagram of ELSA. The system can be divided into three major parts, mechanical part, electrical part, and web server part.

Commented [8]: The description that follows describes what the product (or other deliverable) is and what it does so that the reader knows exactly what is being produced in this project.

Commented [9]: The following information tells the reader how the product/system works.

The mechanical part contains all the components that directly toggle the switch and hold other components. It has two main functions. The first function is to apply force to toggle the switch. To achieve this function, motors and electromagnets will be considered to generate force to push the switch. Also moving parts and a supporting frame is needed, they can be created by using 3D printing technology.

The Electrical part contains the microprocessor and all other ICs to send and receive signals from mechanical parts. Its purpose is to build a control system that is used as an interface to achieve some automation with the webserver. To do so, there will be parts controlling the switch and detecting the status of the switch, and generating signals.

The Web server allows users to configure settings and show data received from the mechanical part. It will use the function from the electrical part to control the switch. This part will involve both front and backend programming.

2.1.3 Components List

Table 1. List of Specifications

Items	Specifications/Model
Microprocessor control board	ESP32 Devkit V1
Gear Motor	GA12-N20 (gear reduction rate 1:250)
H-bridge Module	DRV8833
Tilt Module	GY25z
Shaft	2mm hex shaft
3.6V Battery	SAFT LS14250

2.1.4 ECE Knowledge

ELEC3300 --- Introduction to Embedded Systems:

Most of the knowledge in this course will be used in this project as we are going to use a microprocessor to hold a web server and control the hardware components. We will utilize the general-purpose input/output (GPIO), universal asynchronous receiver-transmitter (UART), and analog to digital converter (ADC) protocol to achieve automation in the product. Also, use interrupts to achieve some manual control.

ELEC3400 --- Introduction to Integrated Circuits and Systems

This course introduces the characteristics of BJT and MOS transistors and shows the principles of building an amplifier circuit for different applications (ADC, DAC, filtering). In this project, the current from the development board is limited and not enough for the motor. Therefore, we are going to build a circuit to provide a stable power supply for all components.

ELEC3120 -- Computer Communication Networks

The network protocol is used in this project to achieve the connection between switches, giving the command, and making a user interface. For example, UDP, HTTP. Understanding them helps create and debug the part where the device needs to communicate with the browser or other devices.

2.2 Objective Statement Execution—Plan for Producing ELSA

In this subsection, we further describe ELSA and outline how we will design, fabricate and program ELSA so that it meets our set objectives.

Commented [10]: This section communicates your plan for completing the entire project. Each objective statement has a plan for how it will be successfully achieved.

2.2.1 Switching mechanism and external device

Objective: To create the switch mechanism of ELSA that fits on a standard existing electrical switch in Hong Kong and can activate the existing switch to turn it on and off as required. The switch mechanism should activate in under 1 second.

In this objective, we produce a functional switching mechanism and make the model for the essential parts. The aim is to create a design that can stick on an existing light switch using double-sided adhesive tapes, and function without anyone holding any components. In the best case, it should be working on two or more brands of a switch, because some of their designs are similar.

For Hong Kong, there are three types of switch, toggle switch, European style rocker switch, and Australian rocker switch, as shown in Fig. 2 (a) – (c). Among them, the European-style rocker switch and Australian rocker switch are used in most residences and buildings.



Figure 2a. Toggle switch



Figure 2b. European style rocker switch



Figure 2c. Australian rocker switch

To simplify the design progress, we selected European-style rocker switches in figure 2b as the basis of the frame design. There are several reasons. First, it uses less force to toggle than the other two types because the moment arm is longer. Second, it has a larger surface area that can apply force, which means it allows a larger tolerance on the design.

Task 1

Aim: Find a way to generate force that pushes the switch on or off.

Expected Outcome: Finding one or more ways that can toggle the button and compare them.

Member in charge: All group members

In this task, we will investigate the methods of actuation to move a standard light switch.

Task 2

Aim: Design and fabricate a simple mechanism that uses the same actuation mechanism from task one.

Expected Outcome: Creating a part that is going to apply force to the switch. Hold the component by hand and measure the distance of the switch and the part that works.

Member in charge: CHAN Siu Ming (Designing and printing), CHAN Tai Man (testing, feedback, and measuring)

In this task, we will use 3D modeling software to design a switching mechanism to apply force to the switch. Then, we will use a 3D printer to fabricate the switch parts and assemble them.

Task 3

Aim: Build a frame to hold all the parts.

Expected Outcome: The frame should be able to stick on a switch and hold all the components from Task 2 in place.

Member in charge: CHAN Siu Ming

In this task, we design the frame and fabricate the frame using a 3D printer.**Task 4**

Aim: To test the mechanism with frame

Expected Outcome: The mechanism should be able to work by connecting to the power manually.

Member in charge: CHAN Tai Man

In this task, we stick the frame and mechanism to a standard switch and apply a current to actuate the switch to see if ELSA works.

Task 5

Aim: Add support to operate with Australian rocker switch

Expected Outcome: Same as task 3, but with the Australian rocker switch.

Member in charge: CHAN Siu Ming

In this task, we design and 3D print a different frame to fit on a different type of standard switch.

2.2.2 Set up Hardware and Software

Objective: To program the microprocessor to link ELSA to a web server so that ELSA can be accurately controlled through a web browser.

In this objective, we will build the hardware and software control systems for ELSA. The aim is to allow ELSA to connect to the web server for handling user requests, report switch status, and control the mechanical part to complete on/off requests accurately with the aid of sensors.

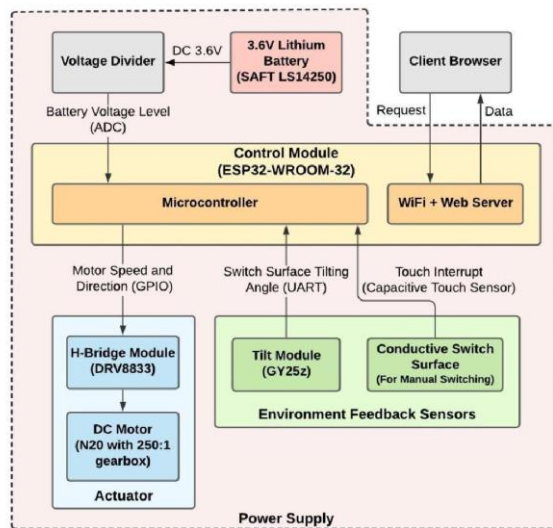


Figure 10. Overview of ELSA Hardware System

Figure 10 shows a detailed design of the hardware part with selected components. It consists of a control module, actuator, environment feedback sensors, and power supply.

We use the ESP32 DEVKIT V1 development board as the control module. It integrates the WiFi module and microcontroller (MCU). While running a web server that handles client requests, it controls the actuator to toggle the switch and receives feedback sensors' signals.

The actuator part holds and rotates the 3D printed moving parts. It includes a DC Motor N20 and H-bridge motor driver DRV8833, which has the following significant features:

- Can power low voltage motors (2.7V - 11.8V) which fits with N20
- Has high energy efficiency

- Provides low power sleep mode so can save power when the switch is in standby mode
- Includes internal shutdown function to protect from short circuit, overcurrent, overheat
- Small and no need heat sink

There are two kinds of environment feedback sensors. The first one is the tilt module, which reports the tilt angle of 3D printed moving parts to the MCU so it can control the actuator rotating the moving parts to the targeted on/off position and know its state when needed.

The second sensor is the capacitive touch sensor in the ESP32 board. When connecting a conductive material such as copper pads to a touch pin of the ESP32 board, the touch sensor will read the capacitance and obtain an analog reading, touching the material will change its reading. Utilizing this technique users can turn the switch on and off manually without using the web browser, when they touch the copper surface on the 3D printed moving part, a drop of capacitance can form a touch interrupt to inform the MCU corresponding on/off request.

For the power supply, we use the SAFT LS14250 3.6V lithium battery. It is small in size (½ AA-size) but has a large capacity (1.2Ah), it occupies less space but can operate for a long time. The battery voltage level is monitored by reading the analog value in a voltage divider using ADC (analog to digital convertor).

Task 1

Aim: Build an experimental hardware setup of figure 10 for program testing

Expected Outcome: A complete prototype including electronic circuit and mechanical parts

Member in charge: CHAN Tai Man

In this task, we will use a breadboard and Dupont wires to construct a prototype circuit based on Figure 10..

Task 2

Before defining the main functions for on/off requests, we need to define sensor functions to let the MCU obtain environment feedback first. In this task, we will define get roll angle functions. Since the gy25z rotates around the x-axis only, we only need to obtain the roll angle. We also define the touch read and touch interrupt functions.

Aim: Define GY25z and touch sensor functions and variables

Expected Outcome: Able to let ESP32 board obtain the roll angle of GY25z when needed, able to apply touch interruptions

Member in charge: CHAN Tai Man

In this task, we will write the code for the touch functions.

Task 3

Aim: Define main functions for ELSA's configurations and routine

Expected Outcome: Functions that allow ELSA to calibrate on/off positions, perform auto and manual switching.

Member in charge: CHAN Tai Man

In this task, we write the code for ELSE to perform manual and automatic switching.

Task 4

Aim: Create a simple web server with existing WiFi to prove the idea of remote control of ELSA through a web browser.

Expected Outcome: ESP32 board connects to an existing WiFi and generates a webpage, other clients who connected with the same WiFi can access the webpage to on/off the switch

Member in charge: CHAN Tai Man

In this task, we will use Arduino sample code to create a simple web server that can handle on/off requests.

2.2.3 Set up a configuration system and implement other features

Objective: To set up a configuration system, implement other features, and create a user interface for ELSA so that ELSA can discover new devices and the user can configure WiFi settings and set automated tasks.

In the second objective, a simple interface will be created with the function of turning on and off the switch. The code in that objective only allows the user to change the configuration in the source code. However, the targeted users are non-technical users. They are not supposed to learn to edit and compile the code themselves. To make ELSA more user-friendly, a way to configure ELSA is needed. Moreover, ELSA should be able to perform more complex tasks, like task scheduling. In this work, we program the microprocessor to link ELSA to a web server so that ELSA can be controlled through a web browser.

A top-down approach is used to design the main structure of the system.

The main working environment should be at home or other places with a stable WiFi network. ELSA does not require an authorization system as the "Sesame Lock" [6] reviewed in the literature review, but it needs to be able to communicate with other switches in the LAN. Therefore, WiFi network access and a method to communicate with other switches are needed.

The targeted users are non-technical users. The web page of ELSA should be accessible from different devices without searching for the IP address. Therefore, multicast-DNS and DNS should be implemented. Also, the web page should be responsive, so different devices can access it with the proper layout.

In this objective, all the code can be classified into two classes, frontend, and backend. The frontend is the user interface, part of them are visual elements, the remaining is event handling and communication with the server. The backend involves the network connection, hosting a web server, and scheduled tasks. Both parts have nothing to do with the circuit. Most of the code should run even without the circuit.

The IDE used is Visual Studio Code with PlatformIO plugin [8]. The first reason for choosing this IDE is because both the frontend language and backend language are supported by the editor, so development can be done without switching between editors. Second, the PlatformIO project is more suitable for programming in a group. It has config file saving configurations for the project. For example, the platform name, board, framework, and library used in the project. Others can load the config and compile the project without extra configuration. If we use ArduinoIDE, all the config needs to be done manually.

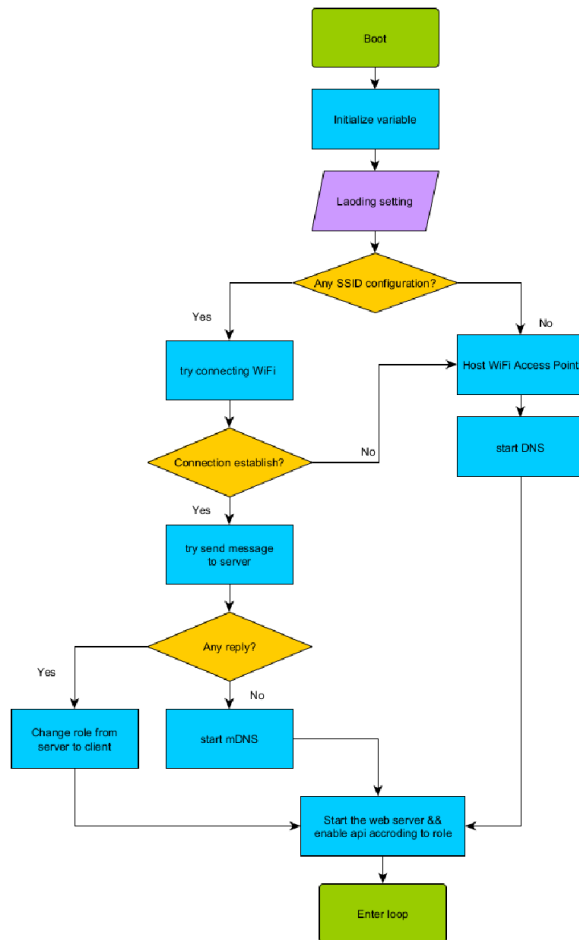


Figure 19. A flowchart showing the rough idea of the backend boot process

The flowchart in figure 19 shows the rough idea about the order of service and the decision needed to be made when ELSA boots up. The tasks in the objective will show the build process of the program according to this flowchart.

Task 1

In the second objective, a web server will be built using TCP. However, the web page content will be hardcoded into the source code. It is hard to read and maintain. This method is suitable for smaller projects that contain few lines of code and doesn't plan to do further development. For this objective, we are going to build more

web page content and API endpoint. Therefore, using a library, which supports reading from files and creating API endpoints, is a better choice.

Aim: Setting up the web server that can handle GET and POST requests.

Expected Outcome: A web server serving a web page and can handle GET and POST requests.

Member in charge: CHAN Siu Ming

The first task is setting up a web server. Building the server in the early stage is because it can help us to debug other parts. Most of the settings in ELSA should not be hardcoded in the source code, such as the WiFi setting. Setting up API in the early stage allows us to perform certain tasks with the "GET" or "POST" command.

Task 2

Also, there should be some kind of tab system to keep content organized instead of placing them on one page.

Aim: Designing a responsive web page that has an extensible tab layout

Expected Outcome: A responsive page with a tab layout.

Member in charge: CHAN Siu Ming

In the second objective, the web server will be built. However, there is no content on the page. In this task, the content will be added to the page. The content of the page needs to fulfill one important requirement, user friendly on both mobile and desktop devices. In other words, the layout of the content should be changed according to different devices.

Task 3

Aim: Connect to WiFi network if there is configuration. Otherwise, host a LAN.

Expected Outcome: Connecting to WiFi if there are any configurations. If it fails to connect or no configuration, host a LAN so the user can connect to the LAN and set configuration.

Member in charge: CHAN Siu Ming

This task will be implemented in the WiFi part. ELSA needs to read data from the configuration and decide to connect to an access point or host an access point.

Task 4

Aim: Use DNS and mDNS to create a domain name that is easier to access

Expected Outcome: Hosting DNS server for access point mode and multicast-DNS for client mode.

Member in charge: CHAN Siu Ming

DNS and multicast DNS should allow all the devices to reach the web page. We will use the library "ESPmDNS.h" from the Arduino framework [9] for the multicast DNS. There will be two functions; one will take the product name as the domain name for multicast DNS, and the other advertises information about network services that the device offers.

The library "DNSServer.h" from the Arduino framework will be used for the DNS. It adds a class called DNSServer. The class can handle the DNS request.

Task 5

Assuming there is more than one device in the same network, then a control system is needed to control each device individually. Otherwise, there will be multiple devices using the same domain name and may cause problems. The target for these tasks is to figure out a way to solve this problem. The problem contains two parts. First, the method to control the device individually.

Second, let the devices discover each other.

Aim: Find a way to control clients individually and set up a discovering system for detecting other switches in the same network.

Expected Outcome: The device can find other clients and users can control each client on the webpage.

Member in charge: CHAN Siu Ming

In this task we will write the code so that when a user accesses the webpage, the page will send a request to get the list on one of the devices. Then using the IP address of the client stored on the main device, the user will get other information like on/off status.

Task 6

Aim: Perform scheduled tasks

Expected Outcome: The devices can perform tasks according to time. Users should be able to configure tasks on the web page.

Member in charge: CHAN Siu Ming, CHAN Tai Man

In this task, we write the code for ELSE to perform scheduled tasks and so the user can get and set the schedule using the webpage.

SECTION 3— Project Planning

3.1 Project Schedule

Table 2. ELSA Schedule.

Objective Statements	Task	Group Member in charge	W K 3 1 1 O c t	W K 4 1 8 O c t	W K 5 2 5 O c t	W K 6 1 5 N o v	W K 7 8 1 N o v	W K 8 1 5 N o v	W K 9 2 2 N o v	W K 1 0 2 N o v	W K 1 1 6 D e c	W K 1 1 2 D e c	W K 1 3 2 D e c	W K 1 4 2 D e c	W K 1 5 3 J a n	W K 1 6 1 J a n	W K 1 7 1 J a n	W K 1 8 4 J a n
Mechanism and frame design																		
	Research	CHAN Tai Man, CHAN Siu Ming																
	Fabricate mechanism	CHAN Tai Man, CHAN Siu Ming																
	Fabricate outer frame for European switch	CHAN Siu Ming																
	Testing the device	CHAN Tai Man																
	Fabricate outer frame for Australian switch	CHAN Tai Man, CHAN Siu Ming																
Set up Hardware and Software																		
	Set up hardware	CHAN Tai Man																
	Define sensor functions	CHAN Tai Man																
	Define major functions	CHAN Tai Man																
	Set up simple web server	CHAN Tai Man																

Continued on the next page

Table 2. ELSA Schedule (Continued)

Objective Statements	Task	Group Member in charge	W K 1 8 2 4 J a n	W K 1 9 3 J a n	W K 2 0 7 F e b	W K 2 1 1 F e b	W K 2 2 2 F e b	W K 2 3 2 F e b	W K 2 4 M a r	W K 2 5 M a r	W K 2 6 M a r	W K 2 7 M a r
Set up Hardware and Software (continued)												
	Set up hardware	CHAN Tai Man										
	Define sensor functions	CHAN Tai Man										
	Define major functions	CHAN Tai Man										
	Set up simple web server	CHAN Tai Man										
Configuration and advanced features												
	Setting up the web server	CHAN Siu Ming										
	Designing a web page	CHAN Siu Ming										
	Connect to WiFi	CHAN Siu Ming										
	Host DNS and multicast-DNS	CHAN Siu Ming										
	Multi-client in LAN	CHAN Tai Man, CHAN Siu Ming										
	Scheduling and configuration of tasks	CHAN Tai Man, CHAN Siu Ming										

3.2 Budget

Table 3. Expected budget.

Items	Cost
ESP-32 Devkit V1 development board	~RMB 26
GA12-N20 (Motor) (gear reduction rate 1:250, with encoder)	~RMB 30
DRV8833 (H-bridge Motor Driver)	~RMB 0.98
GY25z (tilt module)	~RMB 21.5
Hexagon Shaft	HKD 18 (for 4 pieces)
SAFT LS14250 (3.6V Battery)	~RMB 13
Total	RMB 91.48 HKD 18

REFERENCES

- [1] B. Zhou, et al. "Smart home energy management systems: Concept, configurations, and scheduling strategies.", Renewable and Sustainable Energy Reviews, Vol. 61, P.30-40, Aug 2016.
- [2] R. Crist, "Smart bulbs vs. smart switches: The pros and cons of connected lighting", Cnet, Sep 2015, Available: <https://www.cnet.com/news/smart-bulbs-vs-smart-switches-the-pros-and-cons-of-connected-lighting/>
- [3] iDISRUPTED, "How Do Smart Switches Work", Jan 2019, [Online]. Available: <https://idisrupted.com/how-do-smart-light-switches-work/#cheaper-electricity-bills>
- [4] iDISRUPTED, "Smart Bulbs Vs Smart Switches", Nov 2018, [Online]. Available: [https://idisrupted.com/smart-bulbs-vs-smart-switches/#Which To Choose %E2%80%93 Smart Switch, Dimmer, Bulb Or Plug](https://idisrupted.com/smart-bulbs-vs-smart-switches/#Which%20To%20Choose%20Smart%20Switch,%20Dimmer,%20Bulb%20Or%20Plug)
- [5] K. Sandburg, "Smart Lighting & Wireless Future", Strategy Dynamic, Sep 2018, Available: <https://medium.com/strategy-dynamics/smart-lighting-wireless-future-9b67eca73984>
- [6] Candy House, "CandyHouse," [Online]. Available: <https://candyhouse.co/pages/sesame-features>.
- [7] August Home, "Wi-Fi Smart Lock," [Online]. Available: <https://august.com/products/august-wifi-smart-lock>.
- [8] O. K. Cheng and M. C. IP, "FYP - Smart Light Switch", Apr 2021, [Source code]. Available: <https://github.com/JobyCheng/FYP-Smart-Light-Switch>
- [8] PlatformIO Labs, "PlatformIO", [Online]. Available: <https://platformio.org>
- [9] OpenJS Foundation, "jQuery", [Online]. Available: <https://jquery.com>

APPENDICES

Appendix A – Meeting Minutes

Meeting 1

Date: 3/9/2020

Time: 10:30am

Location: Zoom meeting

Attendees: CHAN Tai Man, CHAN Siu Ming

Absent: None

Minutes taken by: CHAN Tai Man

- We all agree to use the laser as the optical communication source
- The main objective of the project is confirmed
- IP suggested to use solar panels as signal receivers and position sensors
- The alignment method is decided (3x3 solar board represent different directions)

Table 1. Action Items from the Previous Meeting

Action Item to be completed	By when	By whom	Status
UOWC Research	Sept 3 rd	All	Completed
Proposal Report introduction	Sep 12 th	CHAN Siu Ming	In progress
Literature review	Sep 12 th	CHAN Tai Man	In progress

Table 2. Action Items for Next Meeting

Action Item to be completed	By when	By whom
Proposal Report objectives 1,2	Sep 14 th	CHAN Siu Ming
Proposal Report objectives 3,4	Sep 14 th	CHAN Tai Man
Finalize the proposal report	Sep 16 th	All

Next Meeting: Oct 9, 15:00, Zoom