

**USING SWARM INTELLIGENCE TECHNIQUE TO OPTIMIZE THE  
ENERGY CONSUMPTION OF SMART HOME APPLICATION  
ARCHITECTURE**

By

Lee Kah Poh

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

**BACHELOR OF COMPUTER SCIENCE (HONOURS)**

Faculty of Information and Communication Technology  
(Kampar Campus)

JUNE 2022

## REPORT STATUS DECLARATION FORM

**Title:** Using Swarm Intelligence Technique To Optimize The Energy Consumption Of Smart Home Application Architecture

**Academic Session:** JUNE 2022

I LEE KAH POH  
(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in  
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.



\_\_\_\_\_  
(Author's signature)



\_\_\_\_\_  
(Supervisor's signature)

**Address:**

32, Hala Bercham Timur 8  
Taman Pakatan  
31400, Ipoh, Perak

Tseu Kwan Lee

\_\_\_\_\_  
Supervisor's name


**Date:** 26/8/2022

**Date:** 2022/09/08



## DECLARATION OF ORIGINALITY

I declare that this report entitled “**USING SWARM INTELLIGENCE TECHNIQUE TO OPTIMIZE THE ENERGY CONSUMPTION OF SMART HOME APPLICATION ARCHITECTURE**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.



Signature : \_\_\_\_\_

Name : Lee Kah Poh

Date : 26/8/2022

## ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Ms. Tseu Kwan Lee who has given me this bright opportunity to engage in a project about Swarm Intelligence based scheduling tasks to optimise energy consumption in IoT hardware. It is my first step to establish a career in applying Swarm Intelligence inside the IoT field. A million thanks to her.

Finally, I must say thanks to my parents and my family for their love, support, and continuous encouragement throughout the course.

## **ABSTRACT**

One of the challenges that Smart Home users confront is optimising the energy consumption of Smart Home devices. The end-user of a Smart Home is unsure when or how to optimise the household's energy consumption efficiently. This situation may increase the country's energy demand and a corresponding acceleration of the greenhouse effect. Additionally, when users run the NSAs unevenly, their energy consumption may surpass the maximum permissible power consumption limit, This situation may lead to a short blackout in Smart Homes. To address such a challenges, hence, this project is to build a mobile application that will enable Smart Home users to control their Smart Home appliances energy consumption effectively. A scheduling task based on Swarm Intelligence will monitor and optimise energy consumption based on human activities. Additionally, this project will examine the notion of Swarm Intelligence in greater detail and the effectiveness with which it achieves the objectives when applied to the scheduling task. Finally, this project will outline the methodology for development, overall flow, system requirements needed, and overall work result.

# TABLE OF CONTENTS

<b>TITLE PAGE</b>	<b>i</b>
<b>REPORT STATUS DECLARATION FORM</b>	<b>ii</b>
<b>FYP THESIS SUBMISSION FORM</b>	<b>iii</b>
<b>DECLARATION OF ORIGINALITY</b>	<b>iv</b>
<b>ACKNOWLEDGEMENTS</b>	<b>v</b>
<b>ABSTRACT</b>	<b>vi</b>
<b>TABLE OF CONTENTS</b>	<b>vii</b>
<b>LIST OF FIGURES</b>	<b>ix</b>
<b>LIST OF TABLES</b>	<b>xi</b>
<b>LIST OF ABBREVIATIONS</b>	<b>xii</b>
<b>CHAPTER 1 INTRODUCTION</b>	<b>1</b>
1.1 Problem Statement	1
1.2 Motivation	2
1.3 Research Objectives	3
1.4 Project Scope and Direction	4
1.5 Contributions	4
1.6 Report Organisation	5
<b>CHAPTER 2 LITERATURE REVIEW</b>	<b>6</b>
2.1 Swarm Intelligences	6
2.2 Previous works on Swarm Intelligence in Optimise Energy Consumption for IoT Application	7
2.3 Particle Swarm Optimization Algorithm	11
2.4 Limitations of Previous Studies	14
<b>CHAPTER 3 PROPOSED METHOD</b>	<b>16</b>
3.1 Application Diagram	16
3.2 System Architecture Diagram	17
3.3 Flowchart	19
3.4 Use Case Diagram	21

3.5	Ackley Function	28
3.6	Pseudocode	29
3.7	Gantt Chart	30
<b>CHAPTER 4 PRELIMINARY WORK</b>		<b>32</b>
4.1	System Components Specifications	32
4.1.1	Hardware	32
4.1.2	Software	36
4.2	Activity Diagram	38
<b>CHAPTER 5 EXPERIMENT/SIMULATION</b>		<b>40</b>
5.1	Setting and Configuration	40
5.1.1	Software Setup	40
5.1.2	Hardware Setup	51
5.2	Coding	52
5.2.1	Arduino IDE	52
5.2.2	PSO Visual Studio Code	57
5.3	System Operation	59
<b>CHAPTER 6 SYSTEM EVALUATION AND DISCUSSION</b>		<b>65</b>
6.1	System Testing and Performance Metrics	65
6.2	Testing Setup and Result	66
6.3	Project Challenges	72
6.4	Objectives Evaluation	73
6.5	Concluding Remark	73
<b>CHAPTER 7 CONCLUSION</b>		<b>75</b>
7.1	Conclusion	75
7.2	Recommendation	76
<b>REFERENCES</b>		<b>77</b>



## LIST OF FIGURES

<b>Figure Number</b>	<b>Title</b>	<b>Page</b>
Figure 2.1.1	Hierarchy of SI-based computation algorithms	6
Figure 2.3.1	Flowchart of the PSO algorithm	13
Figure 3.1.1	Application Diagram	16
Figure 3.2.1	System Architecture Diagram	17
Figure 3.3.1	Flowchart of the pre-development process	19
Figure 3.3.2	Flowchart of the further development process	20
Figure 3.4.1	Use case diagram of the system	21
Figure 3.5.1	Ackley Function	28
Figure 3.5.2	Equation of Ackley Function	28
Figure 3.7.1	Report 1 timeline	30
Figure 3.7.2	Report 2 timeline	31
Figure 4.1.1.a	NodeMCU ESP32 board	33
Figure 4.1.1.b	INA219 DC current sensor	34
Figure 4.1.1.c	Breadboard	34
Figure 4.1.1.d	Different types of jumper wires	35
Figure 4.1.1.e	Load that used in this project	35
Figure 4.1.1.f	4-channel relay module	36
Figure 4.1.1.g	Batteries that attached into the battery holder	36
Figure 4.2.1	Activity diagram of overall system flow	38
Figure 5.1.1.1	Preferences option's location in Arduino IDE	40
Figure 5.1.1.2	Preferences pop out window	41
Figure 5.1.1.3	Location of the Board Manager	41
Figure 5.1.1.4	Board Manager pop out window	42
Figure 5.1.1.5	Location of Manage Libraries	42
Figure 5.1.1.6	Library Manager pop out window	43
Figure 5.1.1.7	Location of the Blynk library	43
Figure 5.1.1.8	Location of the Adafruit INA219 library	44
Figure 5.1.1.9	Create new template pop out window	44

Figure 5.1.1.10	Location of the template ID and device name	45
Figure 5.1.1.11	Location of Datastreams tab	45
Figure 5.1.1.12	Location of web dashboard tab	46
Figure 5.1.1.13	Setting up a new widget	46
Figure 5.1.1.14	Location of the new Blynk IoT application	47
Figure 5.1.1.15	Log in page of Blynk IoT	48
Figure 5.1.1.16	Blynk IoT's home page & add new device page	49
Figure 5.1.1.17	Template of development	50
Figure 5.1.2.1	INA219 after soldering the small parts together	51
Figure 5.1.2.2	Type of precision screwdriver	51
Figure 5.3.1	System circuit for operation	59
Figure 5.3.2	Results obtained from Serial Monitor in switched off condition	60
Figure 5.3.3	Results obtained from the mobile application prototype in switched off condition	61
Figure 5.3.4	Results obtained from Serial Monitor in switched on condition	62
Figure 5.3.5	Results obtained from the mobile application prototype in switched on condition	63
Figure 5.3.6	Results obtained from the AccuBattery	64
Figure 6.2.1	Testing environment setup	66
Figure 6.2.2	Situation 1 final result in five minutes period	67
Figure 6.2.3	Situation 2 final result in five minutes period	67
Figure 6.2.4	Comparison of both situation testing result	69
Figure 6.2.5	AccuBattery application result in Scenario 1	70
Figure 6.2.6	AccuBattery application result in Scenario 2	71

## LIST OF TABLES

<b>Table Number</b>	<b>Title</b>	<b>Page</b>
Table 2.2.1	Electricity bills between unscheduled, BPSO and RSM	10
Table 3.2.1	Pin connection between INA219 and ESP32	17
Table 3.2.2	Pin connection between INA219 and breadboard	18
Table 3.2.3	Pin connection between relay module and ESP32	18
Table 3.4.1	Use case description of register a new account for the system	22
Table 3.4.2	Use case description of access system via smartphone	23
Table 3.4.3	Use case description of view Smart Home appliances statistical data	23
Table 3.4.4	Use case description of control the Smart Home appliances	24
Table 3.4.5	Use case description of manage user	24
Table 3.4.6	Use case description of receive Smart Home appliances statistical data from the Blynk IoT cloud	25
Table 3.4.7	Use case description of apply SI-based scheduling task to the system	26
Table 3.4.8	Use case description of configured information to be displayed in mobile application prototype	26
Table 3.4.9	Use case description of reply Service to Smart Home User	27
Table 4.1.1.1	Specifications of laptop	32
Table 4.1.1.2	Specifications of smart phones	32
Table 6.2.1	Situation 1 results in different period of time	68
Table 6.2.2	Situation 2 results in different period of time	68

## LIST OF ABBREVIATIONS

<i>ABC</i>	Artificial Bee Colony
<i>ACO</i>	Ant Colony Optimization
<i>BFO</i>	Bacterial Foraging Optimization
<i>BFOA</i>	Bacterial Foraging Optimization Algorithm
<i>BOA</i>	Butterfly Optimization Algorithm
<i>BPSO</i>	Binary Particle Swarm Optimization
<i>EB</i>	Electricity Bill
<i>EC</i>	Energy Consumption
<i>FA</i>	Firefly Algorithm
<i>GW</i>	Grey Wolf Optimizer
<i>GWh</i>	Giga watts
<i>HSA</i>	Harmony Search Algorithm
<i>IoT</i>	Internet of Things
<i>IT</i>	Information Technology
<i>NSA</i>	Non-Shiftable Appliances
<i>PAR</i>	Peak-to-Average Ratio
<i>PB</i>	Power Bank
<i>PIO</i>	Pigeon-Inspired Optimization
<i>PSO</i>	Particle Swarm Optimization
<i>PSPSH</i>	Power Scheduling Problem in a Smart Home
<i>RFID</i>	Radio Frequency Identification
<i>RSM</i>	Realistic Scheduling Mechanism
<i>SA</i>	Shiftable-Appliance
<i>SG</i>	Smart Grid
<i>SI</i>	Swarm Intelligence
<i>SSO</i>	Simplified Swarm Optimization
<i>UC</i>	User Comfort
<i>WSN</i>	Wireless Sensor Networks

## CHAPTER 1

### Introduction

The Internet of Things (IoT) is a robust Internet built on advanced technologies such as Radio Frequency Identification (RFID), sensor technology, cloud computing, Wireless Sensor Networks (WSNs), and Smart Home systems [2]. It enables users to connect any devices to the Internet in order to exchange information, communicate, and make decisions.

The development of IoT, however, continues to meet obstacles. For instance, the massive number of devices in the IoT will create issues with energy consumption. According to current energy consumption (EC) research, buildings in Malaysia consume 48 percent of the country's power [1], such that building like commercial structures utilise up to 38,645 Gigawatts (GWh), whereas residential facilities consume 24,709 GWh.

Additionally, because IoT systems are complex and dynamic, some of them may be regarded as NP-Complete problems. The NP-Complete problem refers to any class of computational problems for which no efficient solution algorithm has been found, and the solutions can be hard to compute in a reasonable amount of time [3].

Thus, a technique based on Swarm Intelligence for multi-objective optimisation may be utilised [4]. Swarm intelligence (SI) is an approach for simulating biological group intelligence [5]. Due to the parallelism and distributed nature of SI algorithms, they solve significant non-linear problems with enhanced self-adaptation, resilience, and search capability. In addition, the SI algorithms have utilised heuristic information to get the optimal answer. As a result, it applies to a wide range of optimisation problems (e.g., dynamic optimisation, multi-objective optimisation, and NP problems). With the continued growth of IoT, SI has a bright future in IoT-related applications.

### 1.1 Problem Statement

SI-based algorithms have been used to optimise the EC of Smart Home application architectures in the literature. However, attaining EC through such strategies frequently decreases User Comfort (UC), as these techniques are free to

schedule smart appliances over 24 hours [7]. As a result, Smart Home user is unaware of when or how to optimise the EC efficiently.

This situation may lead to increase electricity bills which may burn a hole in their wallet. Although most Smart Home appliances are means to work 24 hours a day, some Non-Shiftable Appliances (NSAs) such as rice cookers and washing machines are not designed to operate 24 hours a day. The Smart Home user is not required to use these items on a daily or weekly basis or within a specified time frame. When these appliances are used, they will consume more energy than intended.

### **1.2 Motivation**

With the 5G trend continuing to grow, a growing number of households may consider furnishing their lovely home using a Smart Home architecture. For instance, a smart lock is installed on the front entrance, numerous smart CCTV cameras are installed to monitor their home, and smart appliances that operate through voice command are installed. While we can appreciate the convenience of technology, however, we should also consider EC since most smart appliances work 24 hours a day, seven days per week.

Furthermore, from 1990 to 2016, Peninsular Malaysia generated more than 90% of its electricity from fossil fuels. In 2016, for example, coal accounted for up to 52% of energy generation, while gas accounted for 44%. [8]. When Smart Homes require additional electricity to operate the smart appliances, this resulted in an increase in energy demand and thus an increase in fossil fuel consumption to satisfy the needs. Excessive use of fossil fuels to generate power accelerates the rate of global warming.

Moreover, owing to the epidemic in Malaysia, many workers are required to stay at home and do their employer-assigned job. Due to the recent high temperatures in Malaysia, some workers are forced to turn on their air conditioners more frequently in order to maintain a comfortable working atmosphere. Each worker's household's EC might dramatically increase due to the increased usage of equipment and home appliances.

Therefore, the proposed technique in this study is to establish a green IoT with the purpose to minimise EC and greenhouse gas emissions caused by smart home design while retaining the benefits of smart equipment.

### 1.3 Research Objectives

The research objective is significant since it may serve as a guideline on how the project going to be success. The research objective should not be over scope to prevent failure of the project development. Furthermore, the research objective should be expressed in a concise and precise style in order to enable the reader to have a better comprehend of what this project going to deliver.

The first research objective is to study and understand the existing sources of applying SI algorithms in Smart Home scheduling task applications. Followed the second objective is to propose an SI-based scheduling task in optimising the Smart Home appliance's EC. This aims to eliminate user frustration by automating the operation of Smart Home equipment as well as optimising the EC in a more effective approach.

The third objective is to develop a SI-based scheduling task mobile application prototype for Smart Home users. This is because most users is having a smart phone in assisting them in their daily works. Thus, a mobile application is created in order to allow the user to more simple in accessing the smart appliances. The last objective of this project is to evaluate the performance of the proposed method in optimising the EC. This activity is to acquire a better insight into how the SI algorithm effect the performance of the conventional scheduling job.

The research objectives of this project are simplified as follows:

- 1) To study the pattern of energy consumption when applying swarm intelligence algorithm in Smart Home scheduling task application.
- 2) To propose a Swarm Intelligence-based scheduling task in optimising the Smart Home appliances energy consumption.
- 3) To develop a Swarm Intelligence-based scheduling task mobile application prototype for Smart Home users.
- 4) To evaluate the performance of the proposed method in optimising the energy consumption.

## 1.4 Project Scope and Direction

The aim of this project is to develop a mobile application prototype for Smart Homes end-users. The prototype will assist the users in monitoring the EC of their Smart Home appliances. The SI-based algorithm was utilised to simulate a real-time scheduling task to alleviate Smart Home user annoyance.

Besides, this project primarily is coded in the C++ programming language. Additionally, the SI-based scheduling task will be carried out in the combination of C++ programming language and Python programming language. The SI algorithm that would be used is PSO algorithm. Following that, the Android platform will be utilized to evaluate the mobile application prototype.

This project will mainly focus on the DC electrical appliances of the Smart Home, instead of the AC electrical appliances. This is because there is a too wide range of the Smart Homes appliances, in terms of SA, NSA. In addition, the Smart Appliances are mostly costly, and buying one of those would be difficult for a student.

This project as well, will not construct the mobile application from scratch. This is because this project's is primarily focused on proving the Swarm Intelligence concept which can be applied in the task scheduling to reduce EC, not to produce a complete functional mobile application. Hence, this project will utilise the IoT platform-based application to accomplish the creation of the mobile application prototype.

## 1.5 Contributions

Every human being should have start to consider in addressing the issue of excessive EC to mitigate the greenhouse effect. As a result, the EC of the Smart Home user should be monitored regularly via a scheduled task. To further improve the performance, the daily scheduling task based on SI is used to achieve a better level of EC. If all Smart Home users are aware of the benefits of monitoring and optimising their energy used, the proposed technique has the potential to minimise the greenhouse gas emissions produced by Smart Home designs.

Many individuals in the twenty-first century have relied on their smart phones to obtain whatever information they required. Hence, an information platform for user is needed to understand their energy usage through the mobile application. To further



facilitate Smart Home users, this project will design a prototype mobile application that will enable rapid retrieval of information regarding the EC of Smart Home appliances. In addition, this prototype would be able to inspire the software community in order to further continue the work or develop a new project based on this project ideas.

### **1.6 Report Organisation**

The details of this study are shown in the following chapters. In Chapter 2, several related backgrounds such as task scheduling and SI-based algorithm in optimising the EC are discussed. Then, a preliminary study of hardware acquisition and system architecture diagram is shown in Chapter 3. In addition, the report timeline is shown in the form of Gantt chart.

Besides, Chapter 4 describes the hardware and software used in this development. Following Chapter 5 explains the setting up for the pre-development, in terms of software and hardware side. Chapter 5 will also describe the results received by the system operation. Next, Chapter 6 covers the system testing metric and the testing results for the project. Any project challenges faced would be discussed and the objective evaluation would be done after obtaining the testing results. Lastly, Chapter 7 would be the conclusion of this project and some recommendations for future work will also be discussed.

## CHAPTER 2

### Literature Reviews

#### 2.1 Swarm Intelligences

According to the literature [5], the SI algorithms were developed to investigate the concepts of simple people producing complex swarm optimization behaviours through collaboration, organisation, information sharing, and learning among individuals in a swarm [9]. SI algorithms were first proposed in the 1990s, and their applications have since been extensively studied and matured, including ACO by [10] and PSO by [11]. SI algorithms proposed between 2000 and 2010 include [12] that introduced Bacterial Foraging Optimization (BFO), [13] that introduced Artificial Fish Swarm Algorithm, [14] that introduced artificial bee colony (ABC) algorithm, and [15] that introduced firefly algorithm (FA). Newer and more good SI algorithms include [16] that introduced pigeon-inspired optimization (PIO) and grey wolf optimizer (GW) by [17] and butterfly optimization algorithm (BOA) was introduced by [18].

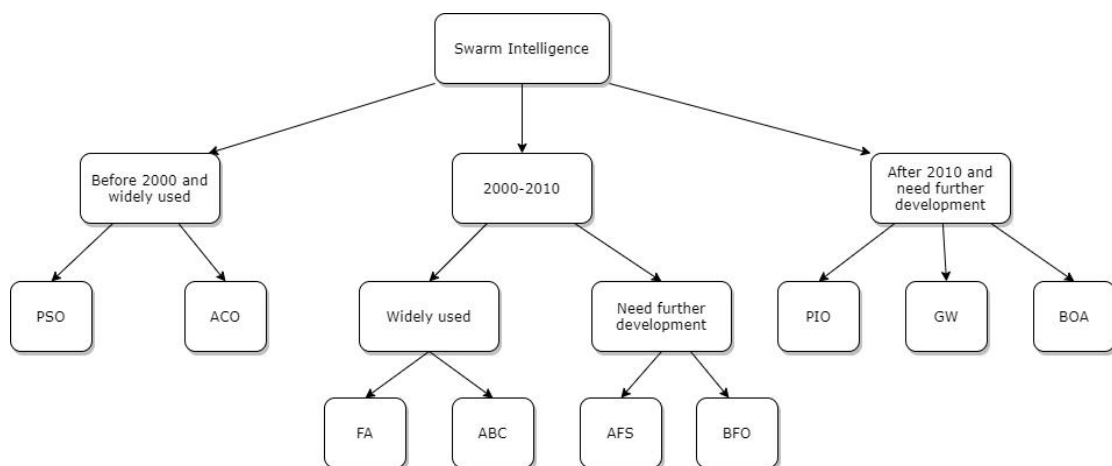


Figure 2.1.1: Hierarchy of SI-based computation algorithms [5]

## 2.2 Previous works on Swarm Intelligence in Optimise Energy Consumption for IoT Application

To begin, many researchers have adopted the Swarm Intelligence Techniques, the nature-inspired heuristic algorithms to optimise the EC in Smart Home appliances. Optimising power consumption for Smart Home appliances in a smart grid is the major problem facing power provider firms, particularly during peak hours, due to the significant impact on a power system's reliability [19].

As a result, power provider firms have implemented dynamic pricing systems that offer varying rates depending on the time horizon, with electricity prices being higher during peak hours owing to increased demand and cheaper during off-peak periods. The Power Scheduling Problem in a Smart Home (PSPSH) is the challenge of scheduling Smart Home appliances at opportune times within a predetermined time range using a dynamic pricing system.

While the authors [7] acknowledge that the task schedules that minimises electricity costs were developed by employing the nature-inspired heuristic algorithms. Besides, they also recognise that by utilising such strategy to accomplish cost minimisation would lead to increase the appliances using period. Additionally, it may result in lower power consumption and cut high amount of electric use during high cost hours without taking appliance utility into consideration, no matter if a nature-inspired heuristic method or a linear/non-linear optimisation model.

However, applying these algorithms leads to one of two outcomes: either the user attempts to control his appliances usage time according to the suggested schedule (lowering the UC level), or he pays higher power costs (EB increment). As a result, this project provided an overview of Smart Grid (SG)s and Smart Homes, as well as their relationship to Power Scheduling Problem in Smart Home (PSPSH) [19].

The researchers in [20] have highlighted that many PSPSH researchers have taken an interest in BFOA and HSA. This is because the BFOA does have flexible restrictions and contain only simple computational equations. In addition, the HSA does feature a powerful and direct search process [21]. On the other hand, the BFOA does exhibits an amazing capacity to exploit a search space. Meanwhile, the HSA has

proved its usefulness in exploring a search space. As a consequence, both algorithms suffer from an unbalanced allocation of exploration and exploitation [21].

On the other side, the PSO has been modified five times to handle PPSH. As a consequence, the PSO is highly resilient and it is a population-based optimisation approach that excels at tackling non-linear and non-differential optimisation problems. Additionally, the PSO is more simple to implement because of the mathematical equations are more clear.

In contrast, the PSO does have a considerable possibility of being stuck in local search. This is because the movement between local and global optima is being restricted by the given parameters. This circumstance occurred notably in a deep and rough search area like PPSH [4].

Notably, several proposed techniques by different researchers disregard NSAs during the scheduling process, resulting in unanticipated power consumption when these appliances are used [23]. As a result, the user's EC may surpass the maximum permissible power consumption limit (threshold). This situation may cause a short blackout in Smart Home. Meanwhile, some proposed methods by different authors grouped the NSAs as the SAs in the scheduling procedure [24]. However, most of the NSAs are manually operated, and no one can predict when or how long people will use them. As a result, specifying time parameters in advance for each NSA appears unfeasible.

On the other hand, most of these nature-inspired heuristic algorithms simply construct a single goal as the goal function, such as the EB reduction. Whereas only a few authors define the goal function using a multi-objective optimisation function [22]. To be explicit, the multi-objective optimisation function refers to reducing EB and boosting the UC level without neglecting the impact of PAR.

Additionally, many of these algorithms only utilise a limited number of Smart Home appliances in a single project. This condition will restrict the examination and assessment of algorithm performance [22].

In light of the preceding, the optimum scenario is to strike a balance between cost-minimisation, the Smart Home appliance that can utility up to the point of user

satisfaction and build a dynamic power consumption limiter that benefits both energy customers and suppliers.

For example, [25] proposed a real-time approach for maximising the energy efficiency of household appliances. One of the algorithm named Binary Particle Swarm Optimization (BPSO) that solves the stated problem by enticing both utilities and customers to participate. The Smart Home appliances are classified into different groups based on their characteristics and the lifestyles of their users. The results demonstrate the proposed scheme's substantial load shifting, energy savings, and cost reduction performance. However, the research did not calculate the PAR value even though they had lowered EB and increased UC level. The PAR is critical for normalising the electric load over a specific period.

Additionally, another researcher has employed the BPSO as a multi-objective optimisation function for scheduling [26]. The aim of this research is to offer an energy-efficient house management system. The aim is to keep power expenses low while meeting operating limitations and minimising disruptions. The proposed technique is made more successful by subdividing the swarm into several sub swarms. The scheduling approach has been demonstrated to be an effective scheme and may offer benefits in scheduling widely diversified and technically complicated interruptible loads.

However, there are several weaknesses in the BPSO approach. To begin, the Realistic Scheduling Mechanism (RSM) was established by few of the researchers to relieve user annoyance and improve the home appliances utilisation by effectively categorising the home appliances according to their constraints and time of usage [7]. Meanwhile, twenty-four hours of a day is separated into four logical sub-time periods, which is 360 minutes or 6 hours each. Only desired appliances (as determined by appliance classification) are scheduled in these sub-time periods. This circumstance may dynamically modelled power usage limiter that includes both the energy user and the electricity supplier by restricting power consumption to enhance appliance utility. The researcher optimally constructed schedules with given limits for each sub-time slot using a the BPSO method. After that, they have done a comparison study between the unscheduled electrical load utilisation, scheduled directly by BPSO and RSM,

expressing user comfort based on cost-effectiveness (reduced EC) and appliance utility. This is to validate the proposed RSM.

Sub-time period	Unscheduled (PKR)	RSM without PB (PKR)	RSM with PB (PKR)	BPSO (PKR)
t1 (24:00 – 06:00)	193	175	68	
t2 (06:00 - 12:00)	130	90	34	
t3 (12:00 - 18:00)	340	130	88	
t4 (18:00 to 24:00)	595	365	265	
T (whole day)	1258	760	498	739

Table 2.2.1: Electricity bills between unscheduled, BPSO and RSM [7]

The difference in electricity bills across the three techniques becomes clear when unscheduled, BPSO, and RSM with and without Power Bank (PB) are compared in Table 2.2.1. In the unscheduled timeslot, the cost is highest, which is 1258PKR. However, the scheduling task using BPSO algorithm has lowered the cost to 739PKR, which is a 42 percent decrease. It is suggested that RSM tends to categorise electrical appliances effectively and gives numerous scheduling assignments with short periods. The RSM give about the same savings as BPSO, as 39 percent as compared to unplanned load utilisation. BPSO beats proposed RSM by 3 percent, which is negligible in light of the UC attained. However, when RSM is paired with PB, the electricity cost is dramatically lowered, saving 65 percent compared to unplanned load utilisation, which is 32 percent more efficient than BPSO.

However, when the researchers combined RSM and PB to increase appliance utilisation and demonstrated that the proposed hybrid technique was more efficient than the BPSO method, the exact figure of PAR reduction was not specified in the study. The research article in [7] previously discussed the precise value of cost-

effectiveness and user comfort. Even if the reader is unaware of the proposed strategy, it has accomplished the purpose of PAR reduction. The RMS identifies and schedules residential electrical appliances, hence lowering electricity costs and increasing appliance utility. However, the percentage drop in total PAR will be more evident if the PAR formula is used to determine the exact figure.

Researchers [27] utilised SI-based algorithms to optimise the EC of distributed systems. The researchers proposed a Simplified Swarm Optimization approach (SSO) and the Dynamic Voltage Scaling for lowering the EC of distributed systems. After then, the SSO has been compared to the Particle Swarm Optimization (PSO) approach. The findings has discovered that the SSO is more efficient, in decreasing the execution time for scheduling and the make span.

Authors in [6] presented a non-invasive load scheduling approach for home user centres based on Particle Swarm Optimization (PSO) (PSO). It is characterized by edge computing, which utilises the processing power of the Internet's edge and IoT data sources to optimise cloud computing technology. The model is based on historical data on energy use and user satisfaction in a Smart Home environment, and information is loaded using the constraint PSO. It analyzes not only the state grid's load dispatching but also local renewable energy sources, which promotes the power grid's safe development.

Besides, researchers in [28] have applied the SI-based algorithms to enhance efficient of smart lighting with suitable light intensity. The researchers suggested that simply adjusting the intensity of the lights generated would be the earlier thought on smart lightning. The energy efficiency may be attained with the capability of dimming and sensing natural light intensity. However, the drawback is the requirement of proper light intensity did not match the expectation on specific time. Hence, by utilizing the PSO algorithm, energy savings reached 67.3 percent while also retaining enough acceptable light intensity.

### **2.3 Particle Swarm Optimization Algorithm**

Researchers Kennedy and Eberhart proposed a SI global random search algorithm named the PSO algorithm. This algorithm simulates migration and swarms behavior during foraging [5]. There are a few rules that need to be followed in the

## CHAPTER 2

flock aggregation model. The first rule is avoiding collisions with neighboring particles, while the particle refers to the potential solution to each optimization problem. The next rule is matching individual velocity in the neighborhood, which is also called an updated personal best. The last rule would be flying to the center of the flock, and the flock as a whole flies to the target, which will allow the particle to converge into a single point to obtain a global optimum. Below is the flowchart about the PSO algorithm:



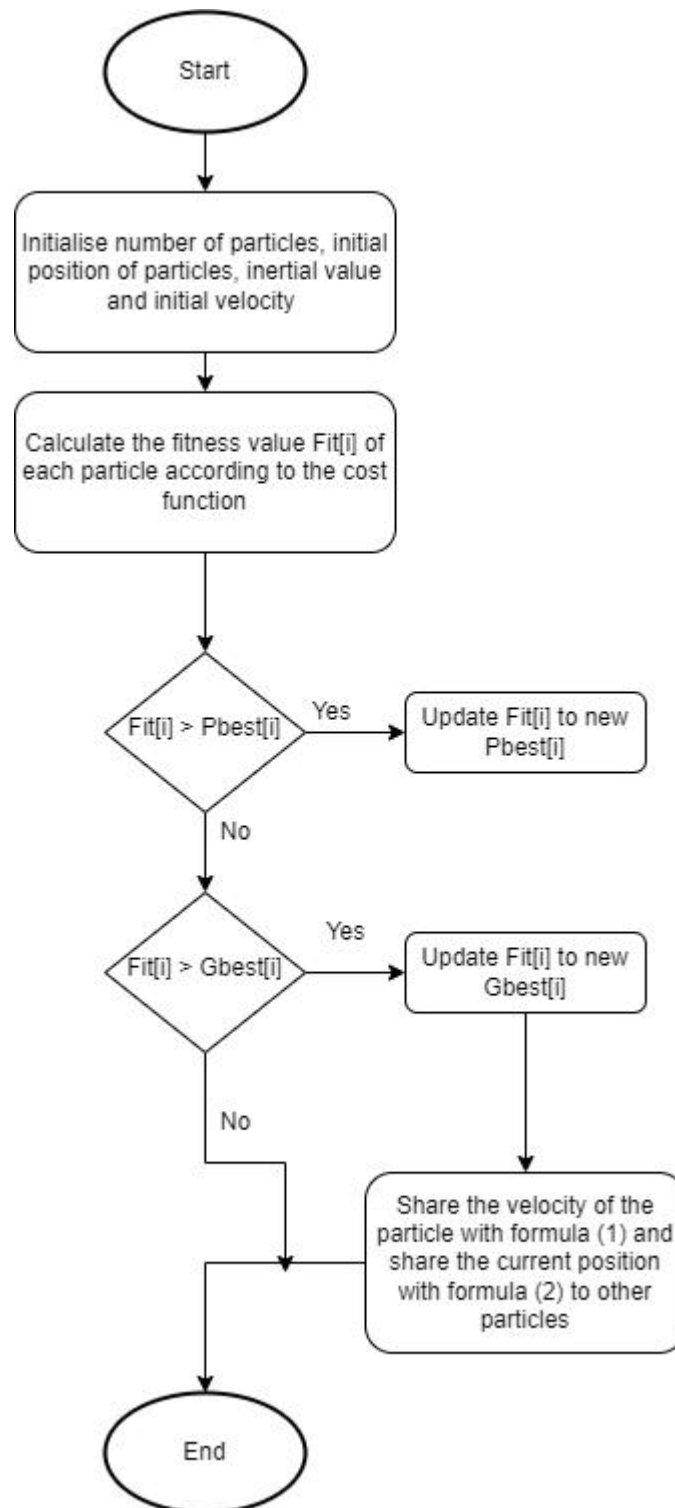


Figure 2.3.1: Flowchart of the PSO algorithm [5]

In figure 2.3.1, when the  $Fit[i]$  in the current iteration was calculated, it was used to compare with the personal best  $Pbest[i]$  indicating the optimal position found by particle  $i$  so far. Besides, the  $Fit[i]$  also used to compare with the  $Gbest[i]$ . The  $Gbest[i]$  refers to the global optimum position found by all particle swarm searches in

a single cost function. If found that either the  $\text{Fit}[i]$  is greater than  $\text{Pbest}[i]$  or the  $\text{Gbest}[i]$ , the particle velocity needs to be updated by using Formula (1) and the particle position needs to be updated by using Formula (2).

$$\text{Formula (1): } V_{id} = \omega V_{id} + C_1 r_1 (\text{Pbest}_i - X_{id}) + C_2 r_2 (\text{Gbest}_i - X_{id})$$

$$\text{Formula (2): } X_{id} = X_{id} + V_{id}$$

The  $\omega$  is the non-negative inertia factor. The smaller the  $\omega$  is, the weaker the global optimisation is, and the local optimisation will be stronger. Besides, the  $C_1$  and  $C_2$  are the acceleration constants, in which  $C_1$  is the individual learning factors that affect each particle and  $C_2$  is the social learning factors that affect each particle.

Authors in [30] mentioned that a recommended choice for learning factors  $c_1$  and  $c_2$  could be the integer 2 since it will makes the weights for social and individual parts to be 1, which is on average. Besides, when  $\omega=1.05$ , the PSO be able to find the global optimum after all iteration. This may be due to the use of the maximum velocity.

#### 2.4 Limitations of Previous Studies

To summarise, multiple proposed methods applying heuristic algorithms by different researchers are disregarding NSAs throughout the scheduling process, resulting in unanticipated power consumption when these appliances are implemented [23]. Furthermore, most of these nature-inspired heuristic algorithms merely generate a single objective as the goal function, such as the EB reduction. Whereas only a few authors define the goal function using a multi-objective optimisation function [22].

However, when Smart Home user contemplates utilising additional NSAs, this conduct may boost the PAR value. To be clear, the PAR is beneficial in balancing power demand and reducing blackout caused by a rapid surge in power usage. Thus, a more extensive investigation may be performed to tackle this problem to design a more effective approach for minimising EB, PAR, and user discomfort while considering NSAs in parallel.

On the other hand, many of these studies only schedule a modest amount number of Smart Home appliances in a single project, restricting the investigation and

assessment of algorithm performance [22]. Additionally, some studies neglected to account for the influence of human activity on the Smart Home appliances EC.

Indeed, a seizable number of Smart Home appliances will operate concurrently in a variety of scenarios. For instance, if the Smart Home user chooses to work from home, the Smart Home appliances usage will increase. In comparison, when the Smart Home user chooses to work outstation, some use of Smart Home appliances such as the Smart TV, air conditioner, and light will be reduced. Thus, future research could take human behaviour into account while optimising the EC of Smart Home application architectures through the Swarm Intelligence technique.

## CHAPTER 3

## Proposed Method

## 3.1 Application Diagram

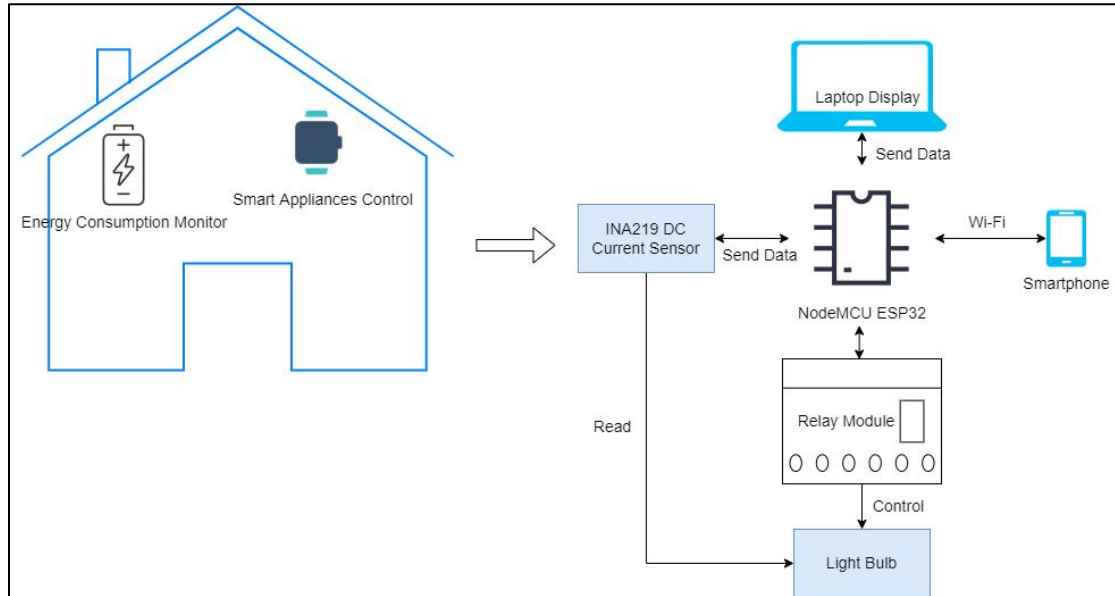


Figure 3.1.1: Application Diagram

Figure 3.1.1 above showed the application diagram of this project. The NodeMCU ESP32 board will be the primary micro-controller to link and synchronise all of the devices to the Blynk IoT Cloud. The laptop was used to upload the code to the NodeMCU ESP32 board. Meanwhile, the NodeMCU ESP32 will receive the monitoring data such as Voltage, Current and Power measurements from the INA219 DC Current Sensor. The data obtained then will transfer to the Blynk IoT Cloud and displayed on the smartphone. Lastly, the NodeMCU ESP32 will send a signal to the Relay Module to toggle on and off of the smart appliances. The switch of transmitting the signal to the Relay Module will be conducted within the smartphone once interconnectivity was done.

### 3.2 System Architecture Diagram

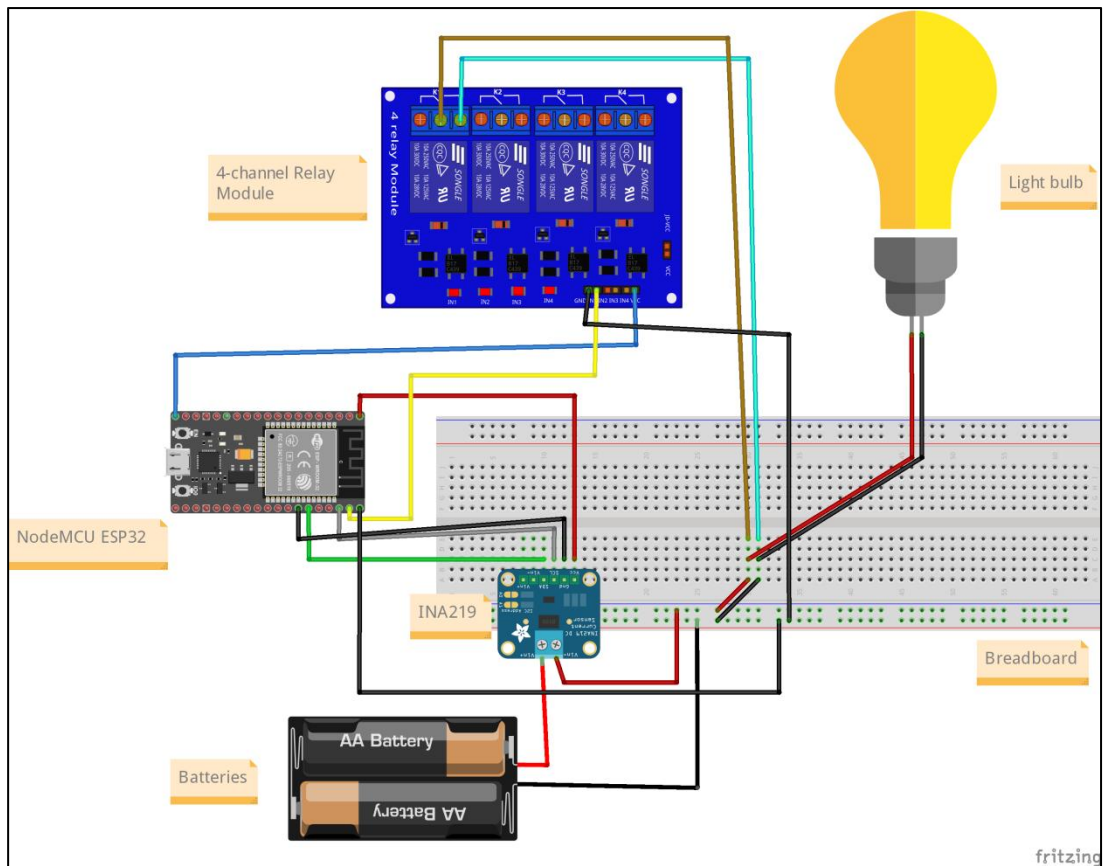


Figure 3.2.1: System Architecture Diagram

#### System pinout connection:

INA219 Pin	ESP32 ESP-WROOM-32
GND	GND
VCC	3V3
SCL	22
SDA	21

Table 3.2.1: Pin connection between INA219 and ESP32

INA219 Pin	Breadboard
Vin-	Positive

V <sub>in</sub> +	Battery positive side
-------------------	-----------------------

Table 3.2.2: Pin connection between INA219 and breadboard

Relay module Pin	ESP32 ESP-WROOM-32
GND	GND
VCC	V <sub>in</sub>
In1	23
COMMON	Load(Positive)
NO	Load(Negative)

Table 3.2.3: Pin connection between relay module and ESP32

### 3.3 Flowchart

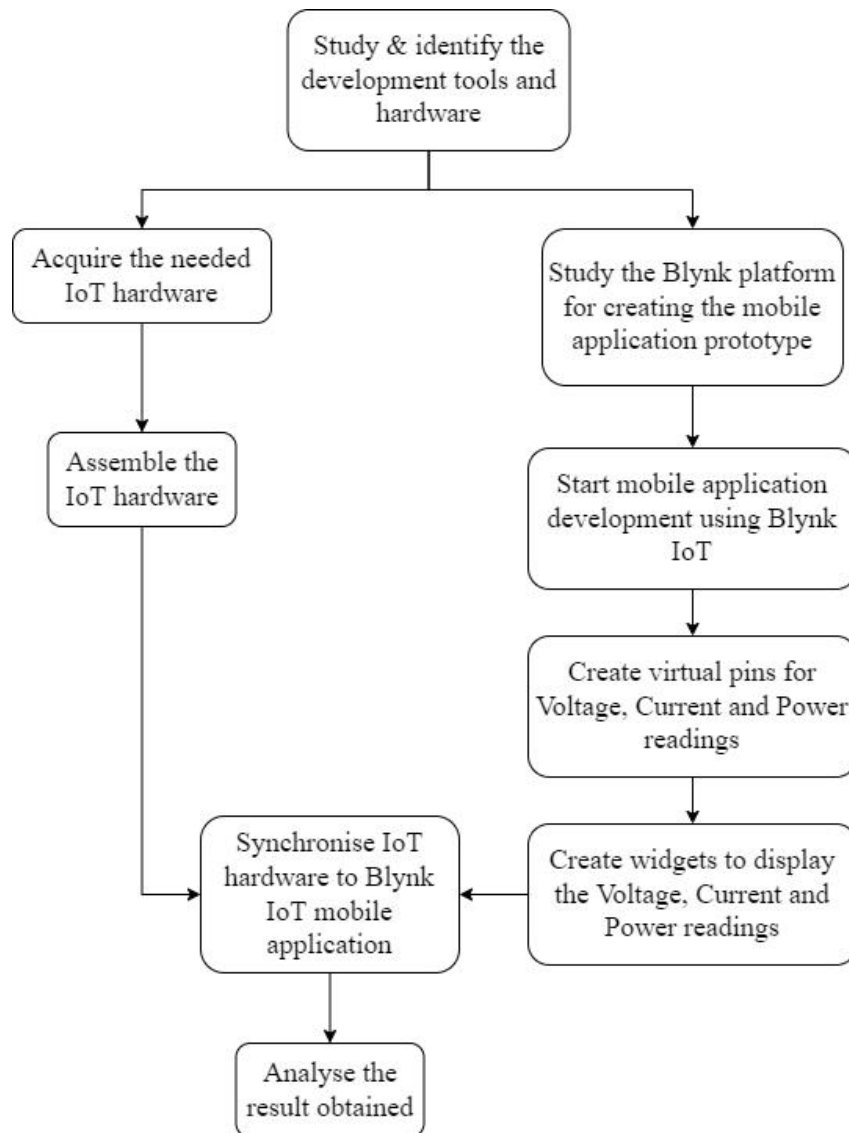


Figure 3.3.1: Flowchart of the pre-development process

Figure 3.3.1 above illustrates the flowchart of the pre-development process. First and foremost, the development tools and hardware that needed for this project were thoroughly researched. Thereby, the Blynk IoT platform was selected in this project for the mobile application prototype development.

Besides, the IoT hardware that needed for this project needs to be examined deeply well. This is because the choice-making for designing the project pre-development circuit is important in order to obtain the AC or DC kind of IoT hardware such as the current sensor and the light bulb. On the other hand, the

connection to be Blynk IoT Cloud and the pre-requisite configuration from the Arduino IDE should be studied thoroughly to avoid connection failure.

After completing assembling the IoT hardware, the creation of the mobile application prototype should be started by configuring the widgets and the virtual pin via the Blynk IoT web application.

After completion of configuration, the synchronisation of the IoT devices and Blynk IoT mobile application should be begun. In this project, the readings such as voltage and current measured by the current sensor need to pass to the Blynk IoT cloud and display them on the Blynk IoT mobile application accordingly. In addition, the light bulb switch should be added too for toggling on and off.

Lastly, the results collected should be analysed attentively to eliminate error readings or specific widgets that did not perform effectively.

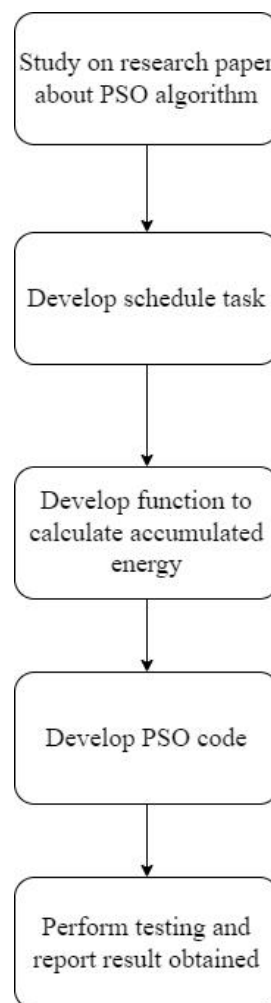


Figure 3.3.2: Flowchart of the further development process



Figure 3.3.2 above illustrates the flowchart of the further development process. First and foremost, the PSO algorithm was studied thoroughly to find a suitable implementation for this project. Meanwhile, a scheduled task should be developed as well because it will be needed before the PSO algorithm has been implemented.

Since the INA219 DC sensor will only monitor the voltage and current flow from the batteries to the light bulb, hence this project needs a function that can calculate the energy left inside the batteries after switching on the light bulb. Next, the PSO algorithm should start building to generate a time value for the scheduled task's timer. When all things are ready, some testing needs to be performed to evaluate the performance of the SI-based scheduled task. The testing results should be recorded accordingly in different periods.

### 3.4 Use Case Diagram

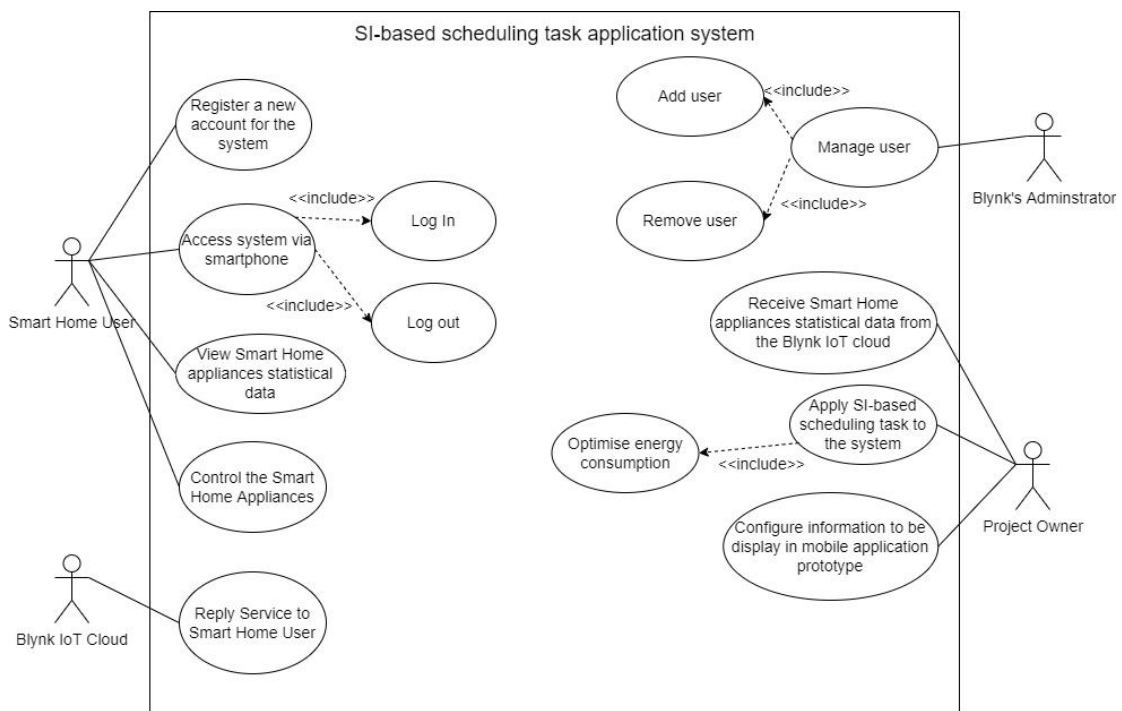


Figure 3.4.1: Use case diagram of the system

Use Case Name	Register a new account for the system
Brief description	User can register a new account to access the system The system will allow the user sign up their new account

Actor	Smart Home User	
Precondition	The Smart Home user does not have any account registered before	
Postcondition	The Smart Home user can sign up for a new account for the application	
Flow of activities	Actor  1. Enter valid credentials format for registering an account.  2. Successfully created a new account.	System  1. Provide a new account for the user.

Table 3.4.1: Use case description of register a new account for the system

Use Case Name	Access system via smartphone	
Brief description	User can access the system using their smartphone  The system will allow the user to log in to their user account	
Actor	Smart Home User	
Precondition	The Smart Home user had registered an account	
Postcondition	The Smart Home user can log in to the application  The Smart Home user can log out to the application	
Flow of activities	Actor  1. Enter the correct credentials to log in to the application.  2. Successfully logged into the application.	System  1. Allow the user to log in to the application.  2. Allow the user to log out from the application.

	3. Allow them log out of their account after clicking the logout button.	
--	--	--

Table 3.4.2: Use case description of access system via smartphone

Use Case Name	View Smart Home appliances statistical data	
Brief description	Users can view the Smart Home appliances energy consumption in real-time	
Actor	Smart Home User	
Precondition	The Smart Home user had logged into the application	
Postcondition	The Smart Home user can view real-time statistical data such as voltage, current and power readings	
Flow of activities	Actor  1. Smart Home user had logged into the application.  2. Able to view the voltage, current, and power readings in real-time.	System  1. Display the statistical data such as voltage, current, and power readings in real-time.

Table 3.4.3: Use case description of view Smart Home appliances statistical data

Use Case Name	Control the Smart Home appliances	
Brief description	Users can control the Smart Home appliances by using the system	
Actor	Smart Home User	
Precondition	The Smart Home user had logged into the application	
Postcondition	The Smart Home user can toggle on and off of the Smart Home appliances	
Flow of activities	Actor	System

	<ol style="list-style-type: none"> <li>1. Smart Home user had logged into the application.</li> <li>2. Able to toggle on and off of the Smart Home appliances at any time.</li> </ol>	<ol style="list-style-type: none"> <li>1. Reply to the service that requested by the Smart Home user.</li> </ol>
--	---	--

Table 3.4.4: Use case description of control the Smart Home appliances

Use Case Name	Manage user	
Brief description	Blynk's administrator can manage the user to allow them to access their own user account	
Actor	Blynk Administrator	
Precondition	The Blynk application has an existing user	
Postcondition	<p>The Blynk administrator can add new user</p> <p>The Blynk administrator can remove an existing user</p>	
Flow of activities	<p>Actor</p> <ol style="list-style-type: none"> <li>1. Allow to add or delete certain Smart Home users.</li> </ol>	<p>System</p> <ol style="list-style-type: none"> <li>1. Allow the changes made by the Blynk Administrator.</li> </ol>

Table 3.4.5: Use case description of manage user

Use Case Name	Receive Smart Home appliances statistical data from the Blynk IoT cloud
Brief description	The project owner can receive the Smart Home appliances statistical data from the Blynk IoT cloud
Actor	Project owner
Precondition	<p>The IoT hardware has been assembled</p> <p>The configuration has been done on Blynk's web application</p>

Postcondition	The project owner can receive Smart Home appliances statistical data such as voltage, current, and power from the Blynk IoT cloud	
Flow of activities	<p>Actor</p> <ol style="list-style-type: none"> <li>1. Assemble the IoT hardware.</li> <li>2. Create the mobile application prototype.</li> <li>3. Perform configuration.</li> <li>4. Receive Smart Home appliances statistical from the Blynk IoT Cloud.</li> </ol>	<p>Blynk IoT cloud</p> <ol style="list-style-type: none"> <li>1. Reply to the request that was made by the Project Owner.</li> </ol>

Table 3.4.6: Use case description of receive Smart Home appliances statistical data from the Blynk IoT cloud

Use Case Name	Apply SI-based scheduling task to the system	
Brief description	The project owner need to optimise the energy consumption of the Smart Home appliances by applying SI-based scheduling task to the system	
Actor	Project Owner	
Precondition	The pre-development has been done	
Postcondition	<p>The project owner can toggle on and off of the light bulb periodically with the help of SI algorithm</p> <p>The project owner can optimise the EC of the Smart Home appliances</p>	
Flow of activities	<p>Actor</p> <ol style="list-style-type: none"> <li>1. The project owner create a scheduling task on top of</li> </ol>	<p>System</p> <ol style="list-style-type: none"> <li>1. Allow the changes made by the project owner.</li> </ol>

	<p>the pre-development work.</p> <p>2. The project owner apply the SI algorithm into the scheduling task.</p> <p>3. The project owner can optimise the EC of the Smart Home appliances.</p>	
--	---	--

Table 3.4.7: Use case description of apply SI-based scheduling task to the system

Use Case Name	Configure information to be displayed in mobile application prototype	
Brief description	The project owner needs to configure certain information to be displayed in mobile application prototype to avoid confusion for the Smart User.	
Actor	Project Owner	
Precondition	The IoT hardware had been assembled	
Postcondition	The project owner can perform configuration to let the voltage, current, and power readings to be visible to the Smart Home user	
Flow of activities	<p>Actor</p> <p>1. Configure the information to be displayed in mobile application prototype.</p> <p>2. Can perform changes at any time.</p>	<p>System</p> <p>1. Allow the changes made by the project owner.</p>

Table 3.4.8: Use case description of configured information to be displayed in mobile application prototype

Use Case Name	Reply Service to Smart Home User
---------------	----------------------------------

Brief description	The Blynk IoT Cloud should be able to reply to the service that requested by the Smart Home User	
Actor	Blynk IoT Cloud	
Precondition	The mobile application prototype had been successfully built	
Postcondition	<p>The Smart Home user can freely view the voltage, current and power consumed by the Smart Home appliances</p> <p>The Smart Home user can freely toggle on and off of the Smart Home appliances</p>	
Flow of activities	<p>Actor</p> <p>1. Blynk IoT Cloud can reply to the service requested.</p>	<p>System</p> <p>1. NA</p>

Table 3.4.9: Use case description of reply Service to Smart Home User

### 3.5 Ackley Function

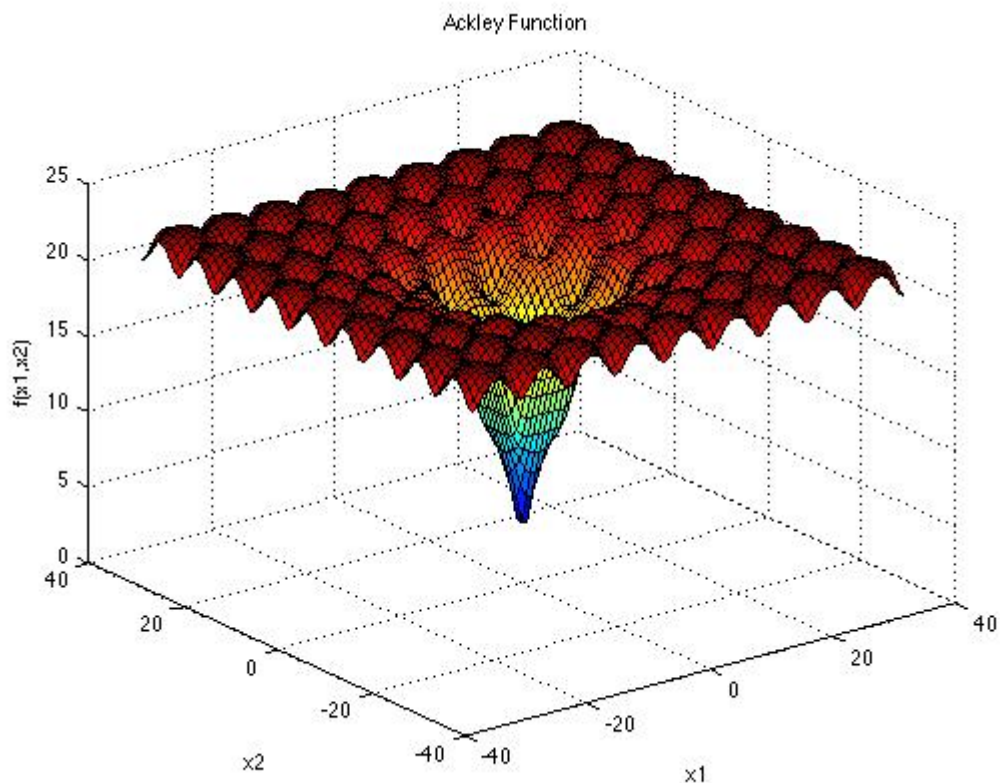


Figure 3.5.1: Ackley Function [31]

It is common practise to test optimisation problems using the Ackley function [31]. By referring Figure 3.5.1, it has a virtually flat outside region and a sizable hole in the middle in two-dimensional form. Optimization problems may run the danger of being stuck in one of the function's numerous local minima.

$$f(\mathbf{x}) = -a \exp \left( -b \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2} \right) - \exp \left( \frac{1}{d} \sum_{i=1}^d \cos(cx_i) \right) + a + \exp(1)$$

Figure 3.5.2: Equation of Ackley Function [31]

Based on [31], the recommended unknown variable values inside the Ackley Function's equation are  $a = 20$ ;  $b = 0.2$ ;  $c = 2\pi$ . Hence, the equation would be converted into the objective function inside the PSO algorithm.



### 3.6 Pseudocode

This program calculates the Most Optimum Value for scheduler timer by using the PSO algorithm

def f(x,y):

    Define objective math function into a computer readable form

Find the global minimum

Initialise hyper-parameter of the algorithm

Create particles with random locations and random velocities inside a region

Initialize data to be used in update function

def update():

    Randomly assign the particle into a region

    Update the velocity by applying the formula (1)

    Update personal best after each iteration

    Update global best after each iteration

Print out the global best after all the particles had converged into a single point

3.7 Gantt Chart

# Report 1 Timeline

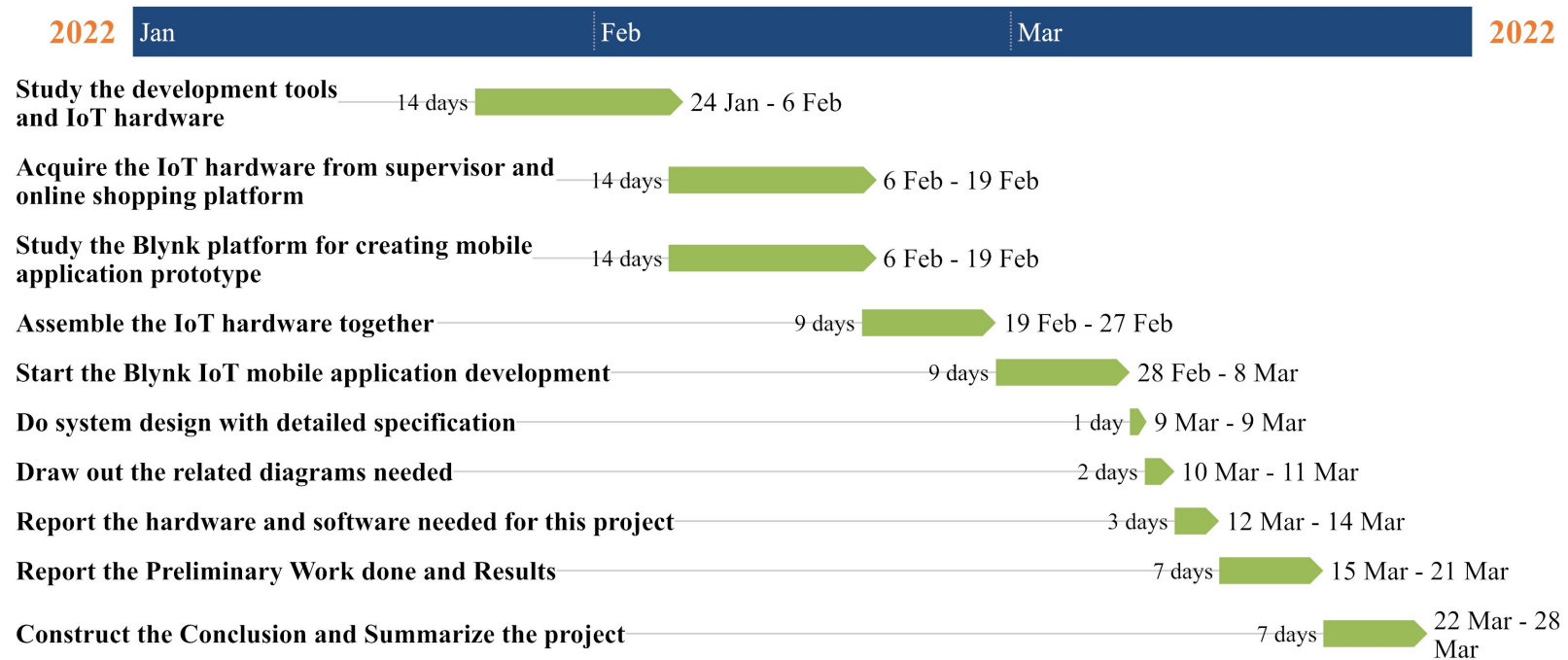


Figure 3.7.1: Report 1 timeline

CHAPTER 3



Figure 3.7.2: Report 2 timeline

## CHAPTER 4

### Preliminary Work

#### 4.1 System Components Specifications

##### 4.1.1 Hardware

This project utilise a laptop as one of the hardware. The laptop is used to execute SI-based scheduling activities and to build the mobile application prototype that monitors the energy usage of Smart Home equipment. Additionally, the laptop is used to execute coding on the Arduino IDE.

As the project's goal is to construct a mobile application for Smart Home customers, therefore a mobile application prototype was created.

Description	Specifications
Model	Lenovo Legion 5
Processor	AMD Ryzen 7 5800H
Operating System	Windows 10
Graphic	NVIDIA GeForce RTX3060
Memory	16GB DDR4 3200 MHz RAM
Storage	1TB SSD

Table 4.1.1.1: Specifications of laptop

Description	Specifications of iOS Smartphone	Specifications of Android Smartphone
Model	iPhone 13 Pro Max	Xiaomi Mi 9
Chipset	Apple A15 Bionic	Qualcomm SM8150 Snapdragon 855 (7 nm)
Operating System	iOS 15	Android 9.0 (Pie)
GPU	Apple GPU (5-core graphics)	Adreno 640
Memory	6GB RAM	6GB RAM
Storage	128GB ROM	128GB ROM

Table 4.1.1.2: Specifications of smart phones

Additionally, this project allow user to monitor EC of Smart Home equipment remotely via a smart phone application. Thus, a NodeMCU ESP32 is required to communicate with the Blynk IoT and current sensor. The following sections discussed the function of various pieces of hardware in detail:

**a. NodeMCU ESP32**

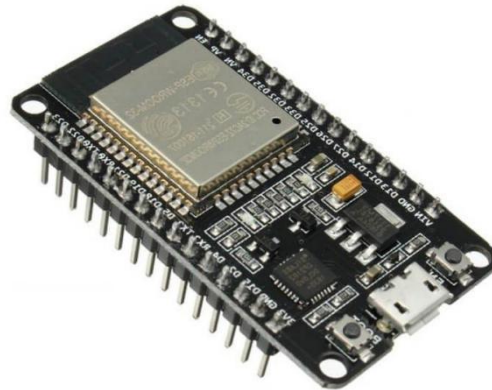


Figure 4.1.1.a: NodeMCU ESP32 board

The NodeMCU ESP32 which acts as the main micro-controller in this project. The NodeMCU is used to process the data acquired from the DC current sensor. The reason why a micro-controller board instead of the Arduino UNO board was used is due to NodeMCU ESP32 has a built-in Wi-Fi module, which allows us to transfer the processed data to the Blynk IoT Cloud. A few settings are required in order to connect the micro-controller board to the Blynk IoT.

**b. INA219 DC Current Sensor**

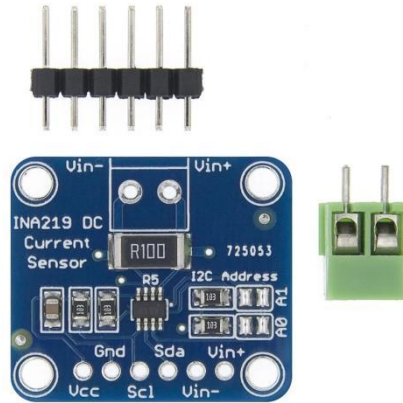


Figure 4.1.1.b: INA219 DC current sensor

The INA219 Current Sensor is a zero drift and bi-directional current or power monitoring module with an I2C interface. Besides, the highest voltage that can be detected with the INA219 sensor is 26V and the maximum current is 3.2A. Therefore, the power consumption can be computed by a simple formula, which is the  $VI = P$ .

### c. Breadboard

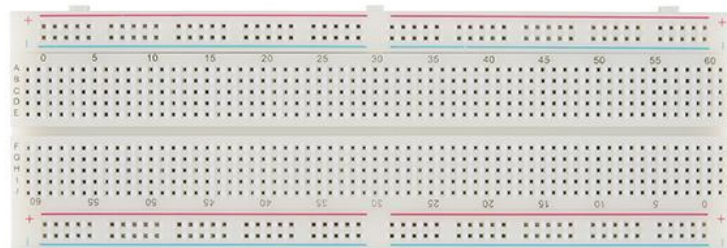


Figure 4.1.1.c: Breadboard

The breadboard enables for quick and easy construction of temporary electrical circuits or to carry out experimentation with circuit design. This project's hardware connection will mostly execute on the breadboard.

### d. Jumper Wires

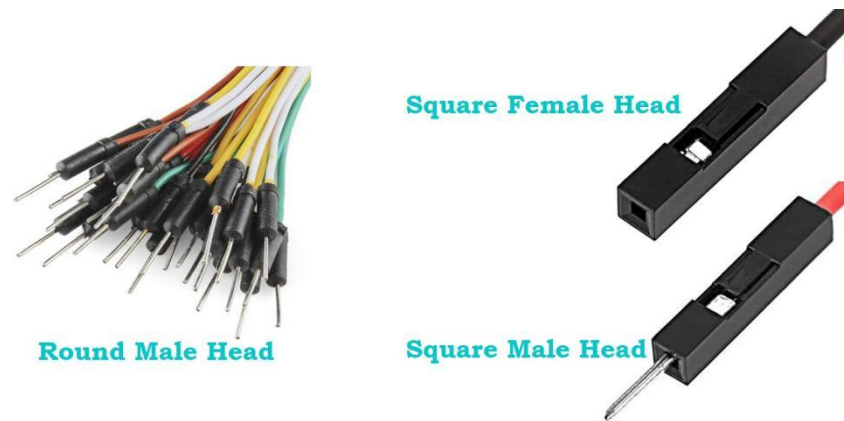


Figure 4.1.1.d: Different types of jumper wires

Jumper wires are simple cables that contain connection pins on either end. These connection pins allow them to be utilised to link two sites without soldering. Jumper wires are mainly applied with the breadboard and other prototyping tools in order to readily adjust a circuit as needed.

**e. Light Bulb Load**



Figure 4.1.1.e: Load that used in this project

The load in this project is intended to simulate the Smart Home environment. Hence, the loads in this project would be the DC light bulb. The light bulb holder need to buy it together too.

**f. 4-channel Relay module**



Figure 4.1.1.f: 4-channel relay module

The power relay module, is also known as an electrical switch controlled by an electromagnet. The electromagnet is operated by a separate low-power pulse from a micro-controller. When it is engaged, the magnetic will start pulling to either open or stop an electrical circuit. Hence, this project will use the 4-channel relay module to turn on and off the light bulb for future task scheduling creation purposes.

#### g. Battery and battery holder

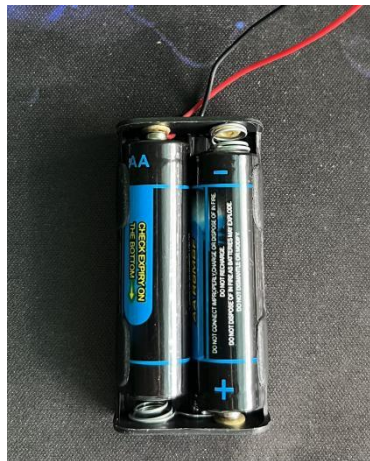


Figure 4.1.1.g: Batteries that attached into the battery holder

This project will use two double A batteries to power up the DC light bulb. A battery holder is needed to connect the batteries to the circuit.

### 4.1.2 Software

The first software used in this project was Arduino IDE, and most of the code: acquiring the current, voltage, and power readings and code for transferring the



sensor's data to the Blynk IoT Cloud; were written on this software. Also, the scheduler task and the function to calculate accumulated energy inside the battery would be created for testing purposes.

The second piece of software employed in this project is Blynk IoT. Both of the mobile-based and web-based Blynk IoT applications were deployed in this project. Firstly, the mobile-based Blynk IoT application would be utilised to develop the mobile application prototype template for this project. The template would include the LCD widget to display the value of current, voltage, and power levels correspondingly.

On the other side, the web-based Blynk IoT application was used to configure the micro-controller device connection. The Blynk IoT application will generate a unique template ID, device name, and authorisation token for us to use in creating the personal design of the development.

Moreover, the AccuBattery application was used as an energy profiler. AccuBattery is a battery monitoring app that allows us to monitor the mobile battery health, real-time charge and discharge information, and real-time tracking of the energy use per application.

Lastly, the Visual Studio Code was used to implement the PSO algorithm based on the objective function that is suitable for this project. The objective function should be converted into a computer-readable form to generate the most optimum time value for the scheduler timer that was built inside the Arduino IDE.

4.2 Activity Diagram

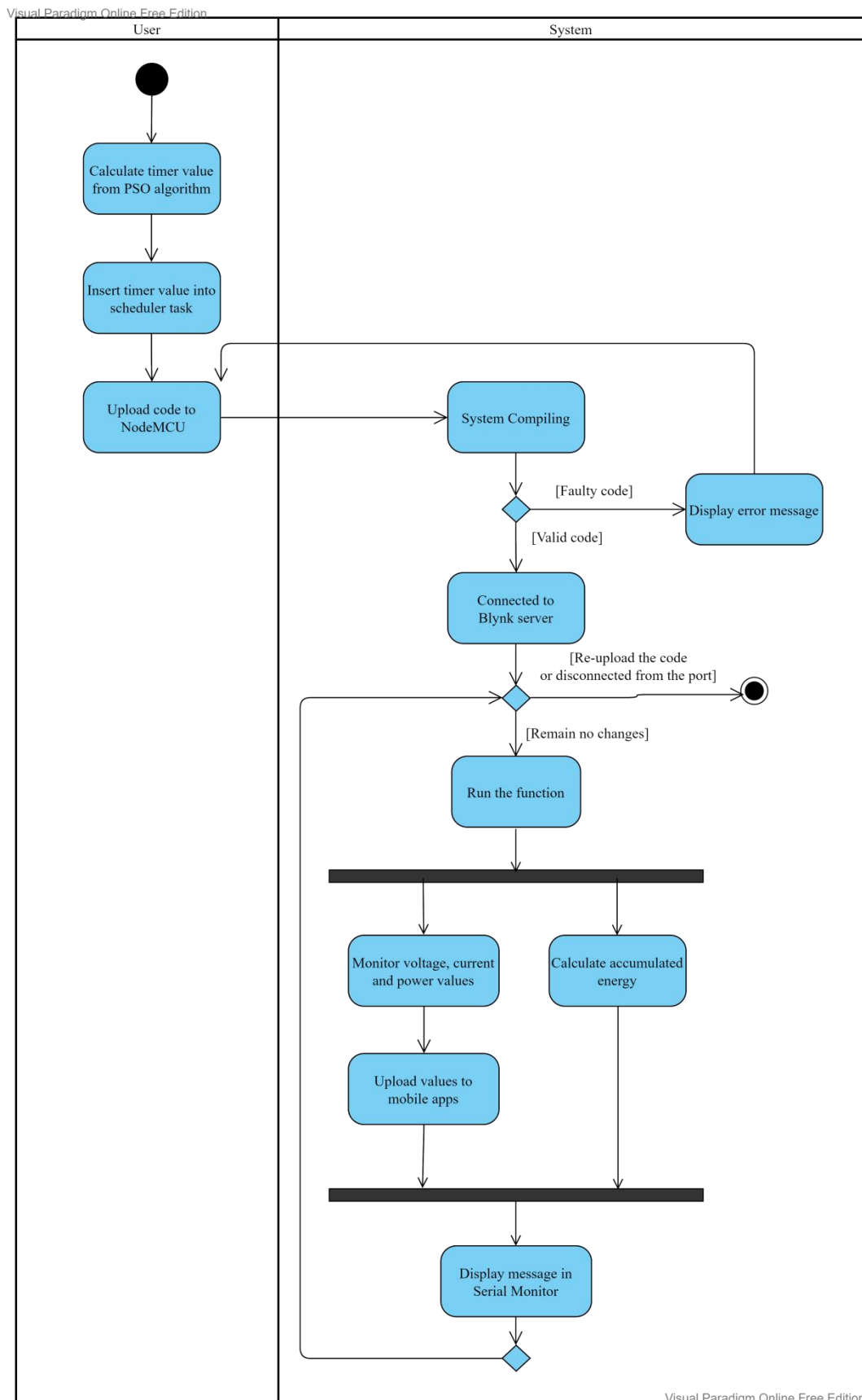


Figure 4.2.1: Activity diagram of overall system flow

The overall system flow was illustrated in Figure 4.2.1. Starting from the user calculate the most optimum timer value from the PSO algorithm. The value calculated would be inserted into the scheduler task as a parameter. After everything were good to go, the code may be uploaded by the user to the NodeMCU ESP32.

The system will start compiling the uploaded code. If found that any error in the code, the system will stop the uploading process immediately. In contrast, if the code is valid, meaning that the system was connected to the Blynk server instantly. Then, by referring “Run the function” activity, all the functions inside the code will start working, such as the monitor voltage, current and power function, calculate accumulated energy function, and so on. After the code had run one round (per second), it will loop back to the beginning of the “Run function” activity. Then the user may decide whether remain no changes or re-upload the code or disconnect the USB port from the PC. The latter action may lead to stop the system from continue.

## CHAPTER 5

### Experiment/Simulation

#### 5.1 Setting and Configuration

##### 5.1.1 Software Setup

This project will utilise several software to accomplish the pre-development. The software needed comprises Arduino IDE, Blynk web application Blynk mobile application, and AccuBattery.

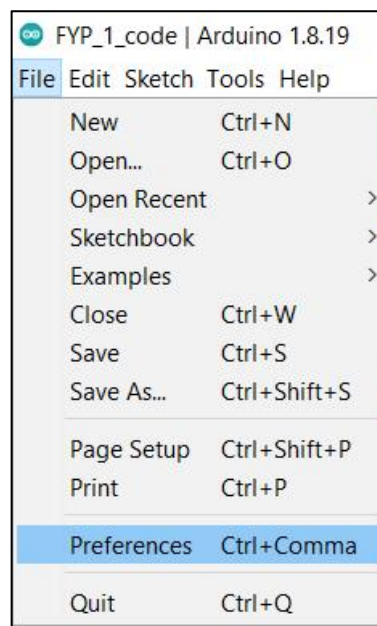


Figure 5.1.1.1: Preferences option's location in Arduino IDE

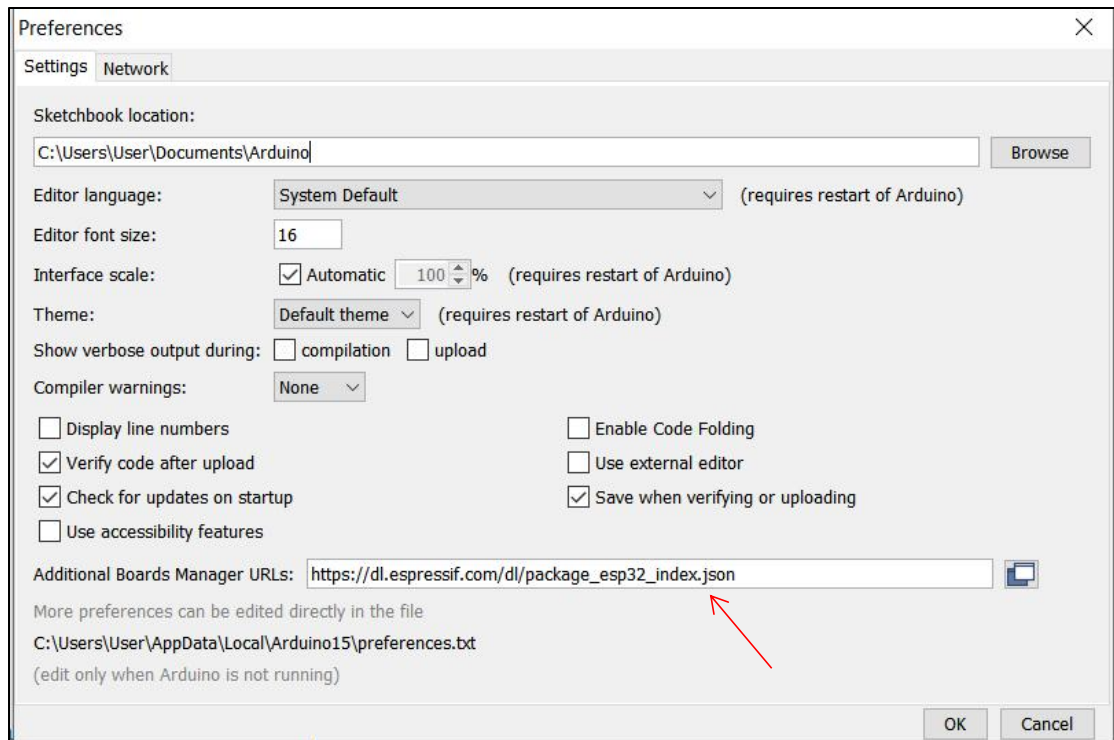


Figure 5.1.1.2: Preferences pop out window

Most of the coding will be written inside the Arduino IDE. After the installation of the Arduino IDE, the first thing that needs to be added is the JSON link of the NodeMCU ESP32 board. The JSON link can be downloaded from the internet at [https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json). After navigating to the “Preferences” based on Figure 5.1.1.1, a window shown in Figure 5.1.1.2 will be popped out. The JSON link was copied and pasted into the text box labeled “Additional Board Manager URLs”.

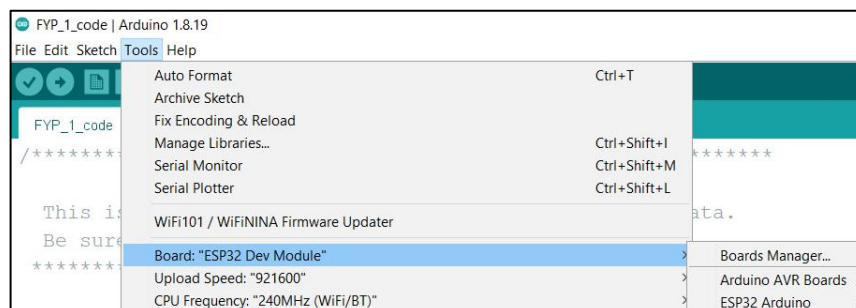


Figure 5.1.1.3: Location of the Board Manager



Figure 5.1.1.4: Board Manager pop out window

The ESP32 board would then be installed inside the Arduino IDE. After navigating to the Boards Manager as shown in Figure 5.1.1.3, a pop-out window shown in Figure 5.1.1.4 will be visible. The word “ESP32” was entered into the search bar. Following that, the available board will be revealed and the board will have been downloaded successfully.

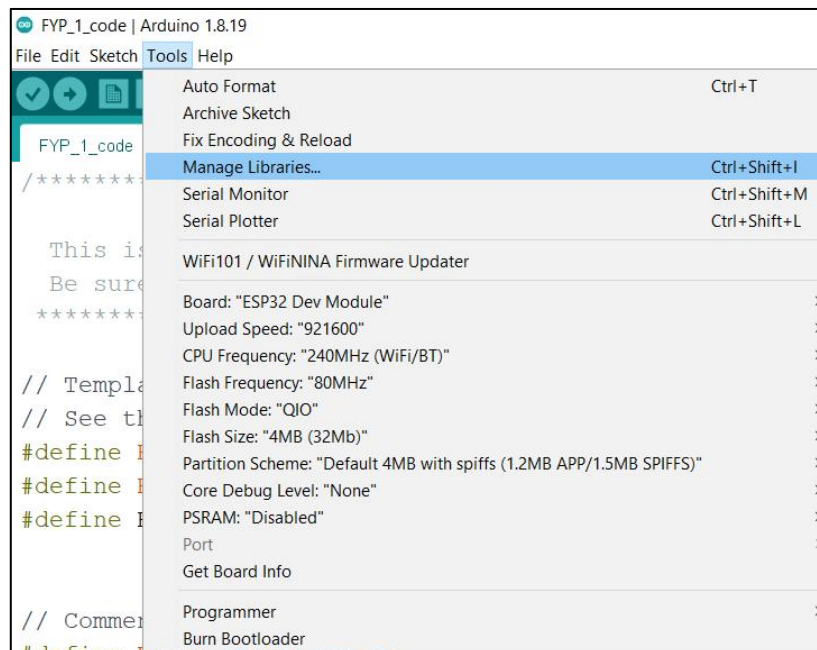


Figure 5.1.1.5: Location of Manage Libraries

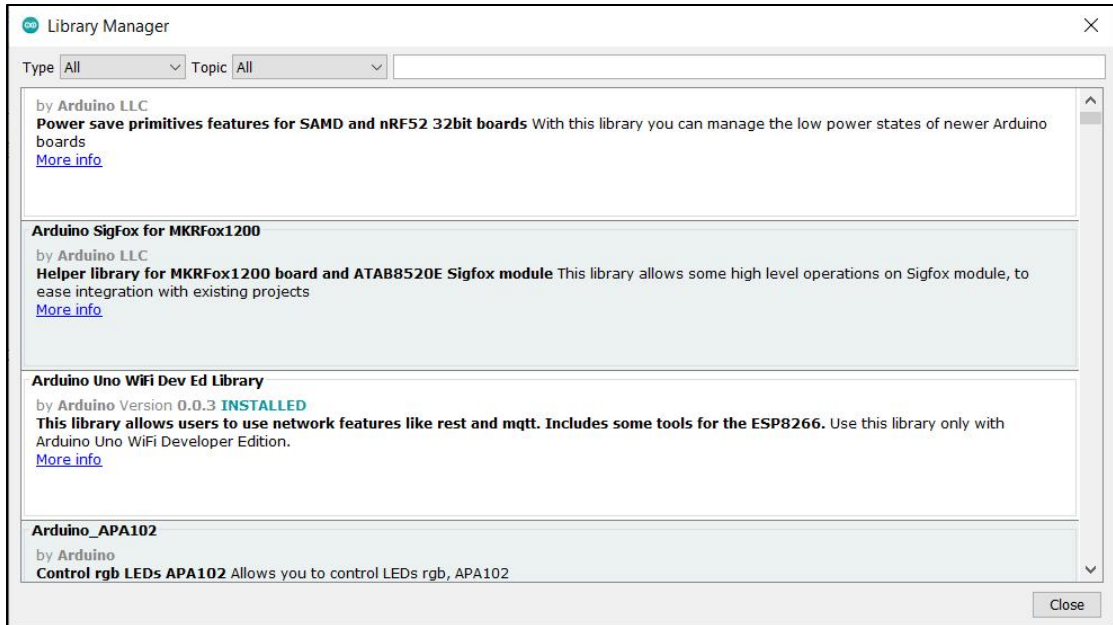


Figure 5.1.1.6: Library Manager pop out window

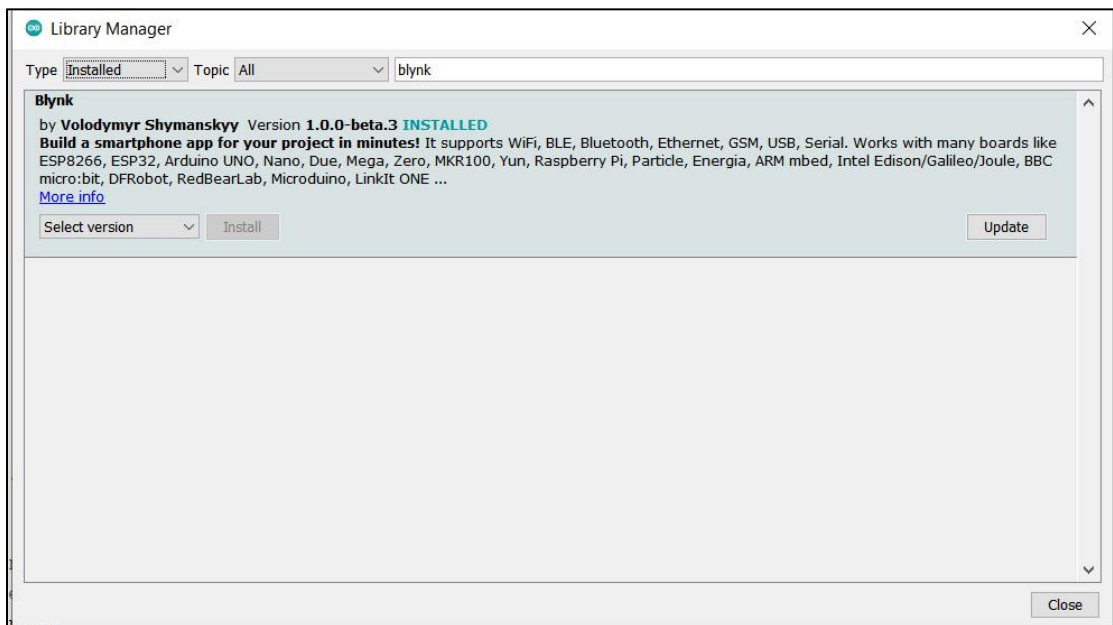


Figure 5.1.1.7: Location of the Blynk library

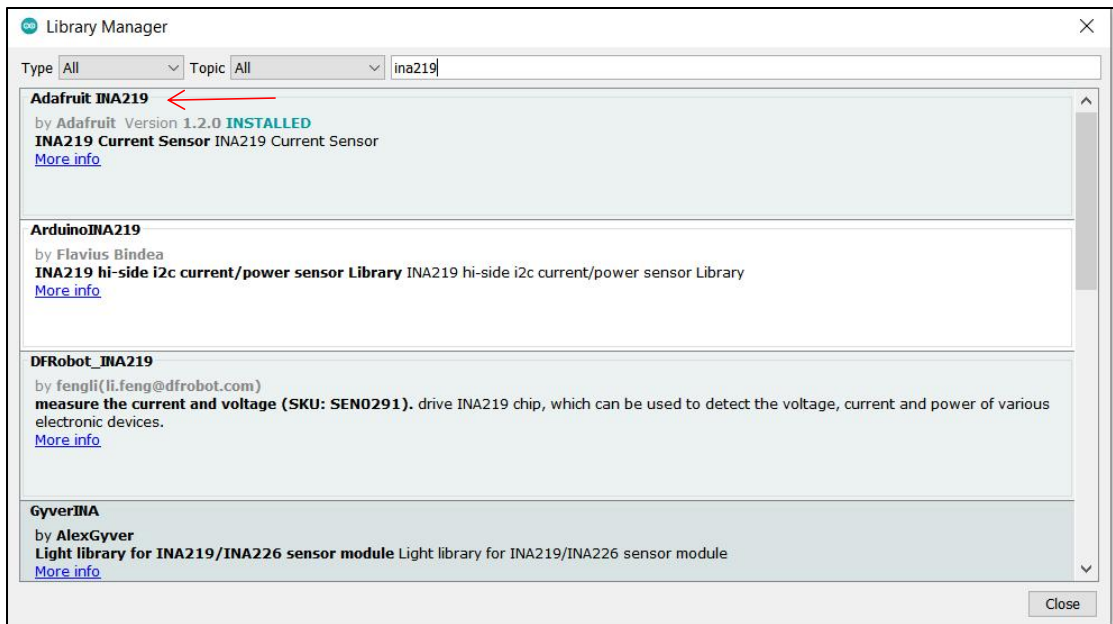


Figure 5.1.1.8: Location of the Adafruit INA219 library

Following that, the installation of the required: Blynk and Adafruit INA219 libraries were installed. The “Manage Libraries” was navigated as depicted in Figure 5.1.1.5. After that, a pop out window shown in Figure 5.1.1.6 were displayed. The “Blynk” phrase was entered into the search bar to search for Blynk library. When the libraries are successfully being downloaded, the status of installed were display as shown in Figure 5.1.1.7. Similar status was shown in Figure 5.1.1.8.

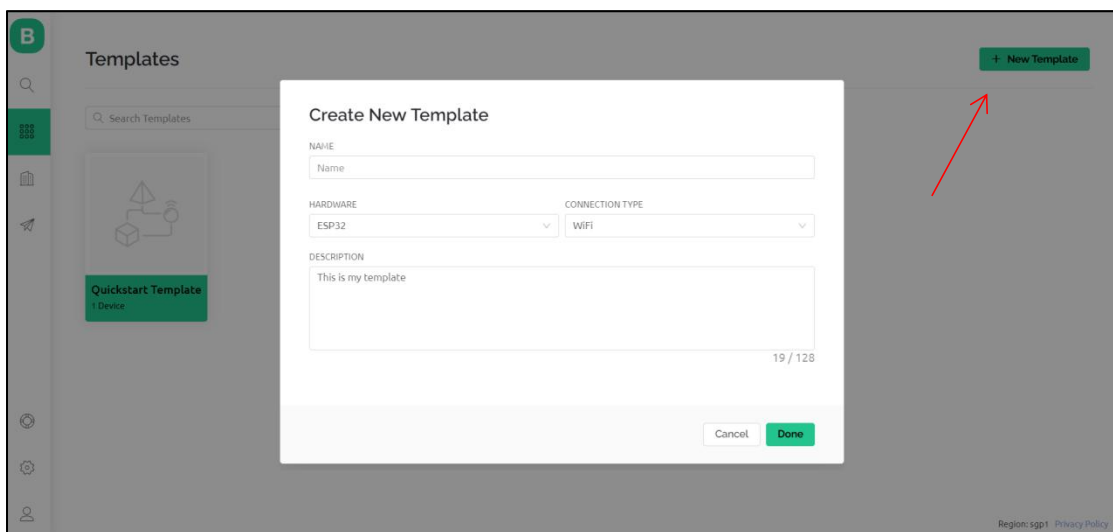


Figure 5.1.1.9: Create new template pop out window



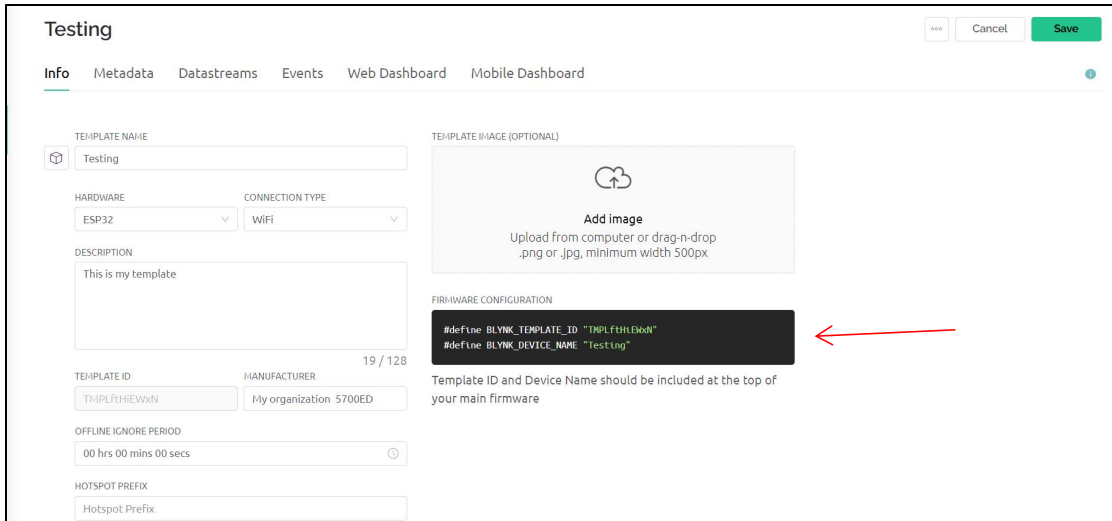


Figure 5.1.1.10: Location of the template ID and device name

After registering an account, a new template was established, which is required before beginning development. By referring the Figure 5.1.1.9, the pop out window should be visible after pressing the “New Template” button. Afterward, the essential data such as name, hardware, and connection type, for the new template were entered. This step will cause the Blynk to automatically generate the template ID and device name for the template that was just created, as shown in Figure 5.1.1.10.

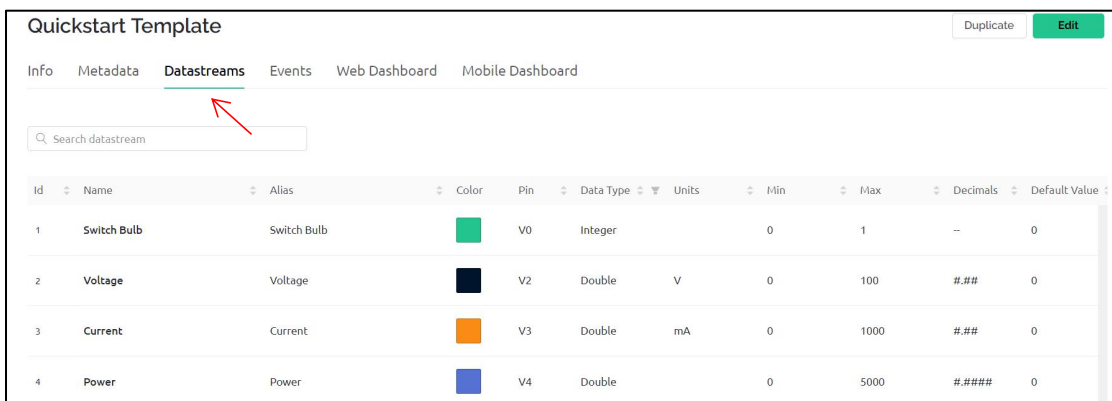


Figure 5.1.1.11: Location of Datastreams tab

Following, the Datastreams tab will allow the user to construct their own virtual pin to serve a distinct function.

Figure 5.1.1.11 displayed above indicates that this project has created some of the virtual pins: V0, V1, V2, V3, and V4. V0 pin was assigned to the light bulb’s switch. The Min and Max values were set to 0 and 1, respectively, with the purpose to toggle on and off of the light bulb. V2 pin was assigned to the Voltage, which is to

receive and show the Voltage readings on the mobile application’s dashboard. V3 pin was assigned to the Current which is responsible for receiving and displaying the Current readings.

Lastly, the V4 pin was assigned to the Power, which is responsible for receiving and displaying the Power values. All three of these pins will change the data type from string to double, in order to display the decimal numbers. In addition, each of the pins have it own designate characteristic colour, units, and the range of decimal digits.

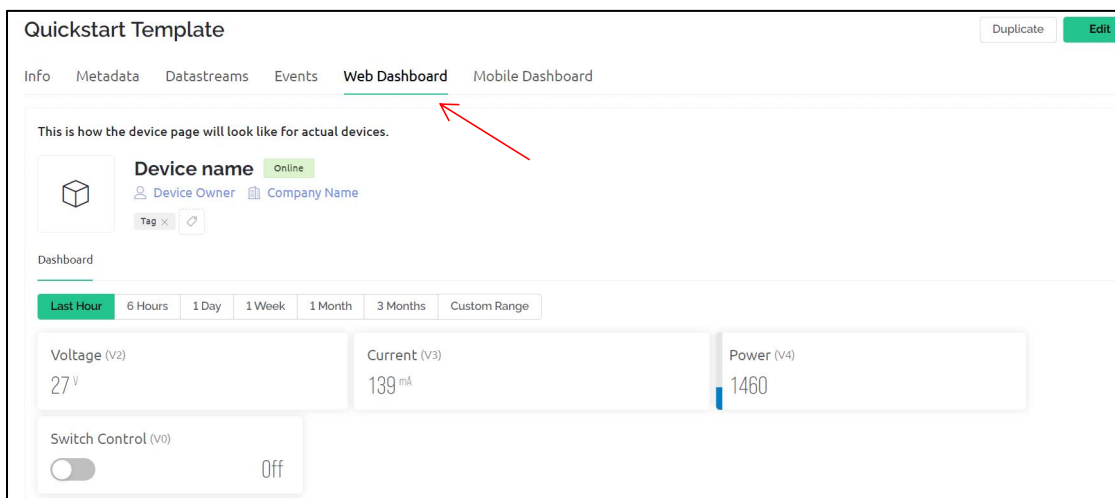


Figure 5.1.1.12: Location of web dashboard tab

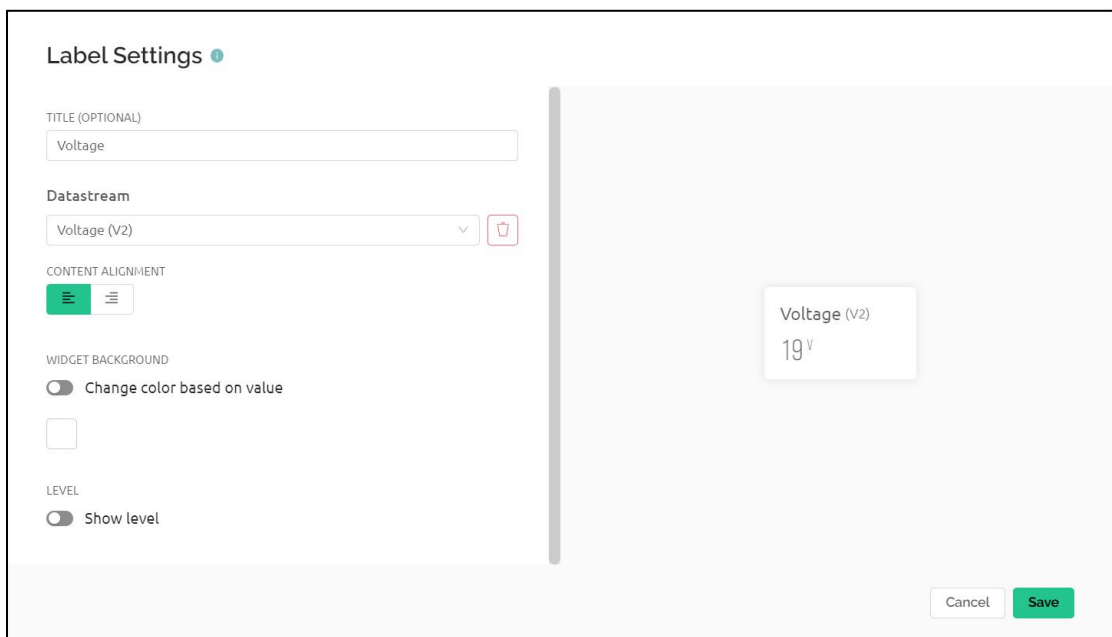


Figure 5.1.1.13: Setting up a new widget

Upon creating all the virtual pins, some settings need to be done within the web dashboard. So, the required widgets were created as seen in Figure 5.1.1.12 above.

Based on Figure 5.1.1.12, the first row of widgets would be the LCD, which would show the Voltage, Current, and Power numbers. The second row of the widgets is a switch control that toggles the light bulb on and off. The setup configuration is quite straightforward; a descriptive title was entered and the suitable Datastream (also known as a virtual pin) that had just been created was selected. The example is presented in Figure 5.1.1.13. The setting should be done after hitting the “save” button.

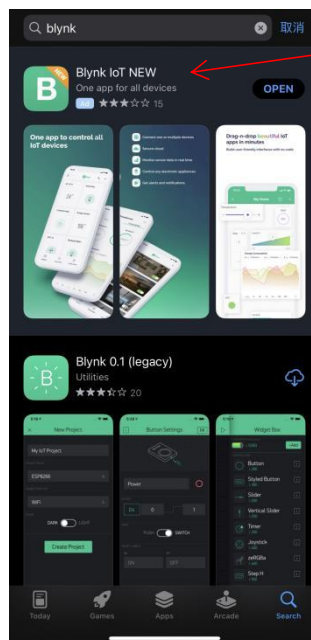


Figure 5.1.1.14: Location of the new Blynk IoT application

After completion of setting the web application, the setup for the mobile application is required also. To begin, the “Blynk IoT NEW” application is needed to download from PlayStore/AppStore as illustrated in Figure 5.1.1.14. This is to demonstrate that both mobile OS platforms can work together, meaning that the development done on the iOS platform can also be synchronised to the Android platform.

Hence, the mobile application development will mostly develop on the iOS platform. Blynk IoT New should be chosen [from the drop-down menu or which

menu]. This is because the Blynk 0.1 (legacy) version does not accept new users and does not support from the developer side, meaning that it will be not accessible by the new user.

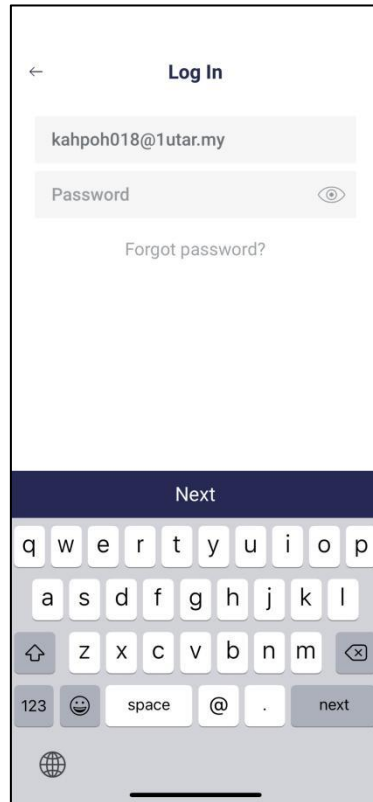


Figure 5.1.1.15: Log in page of Blynk IoT

After completing installing the Blynk IoT mobile application, the app itself is allowed the user to sign up and log in with valid credentials as shown in Figure 5.1.1.15.

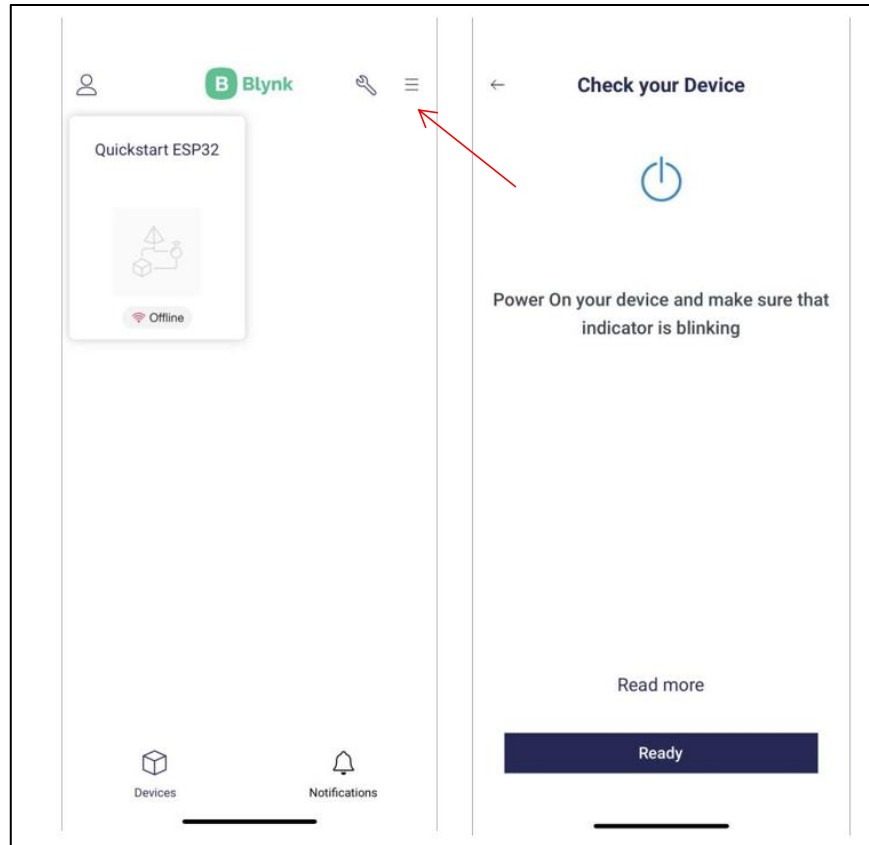


Figure 5.1.1.16: Blynk IoT’s home page & add new device page

After login, the template created from the web application illustrated in Figure 5.1.1.10 should appear as shown in Figure 5.1.1.16 above.

When clicked on the Hamburger menu, the application present “Add New Device” and allow the user to connect their microcontroller board. Figure 5.1.1.1 illustrates the Blynk IoT mobile application provides a step-by-step guide to assist users in connecting their board to the Blynk IoT Cloud.

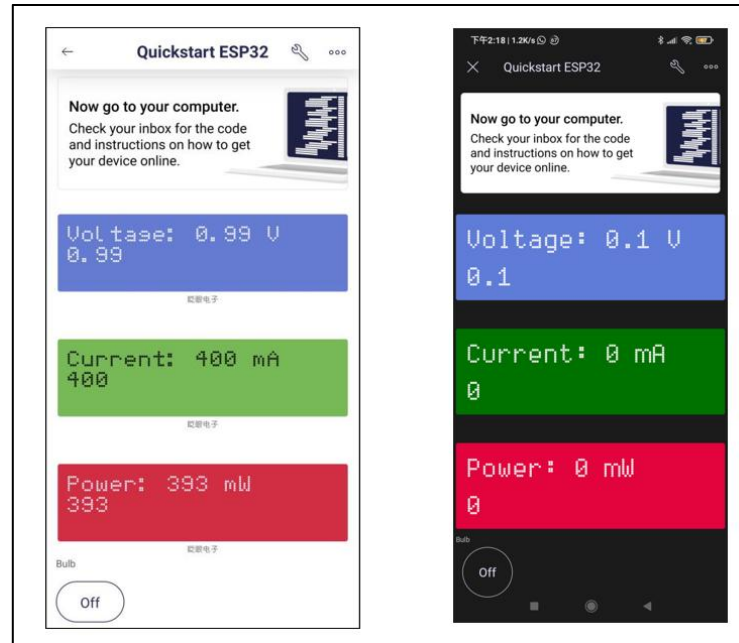


Figure 5.1.1.17: Template of development

After the board was connected to the Blynk IoT Cloud, the necessary widgets for the mobile application as illustrated in Figure 5.1.1.17 should be created. The first three widgets that created in this project were (i) LCD, which is used to display the Voltage, (ii) Current and Power readings. The final widget is the (iii) switch, which is used to toggle on and off the light bulb.

Upon completing the development within the template, the development from the left-hand side (iOS platform) will be synchronised to the right-hand side (Android platform) as illustrated in Figure 5.1.1.17, meaning that the development would be accessible either in the iOS or Android platform.

The AccuBattery, on the other hand, does not require any setup since the application is a free apps which is available to download in the PlayStore.

## 5.1.2 Hardware Setup

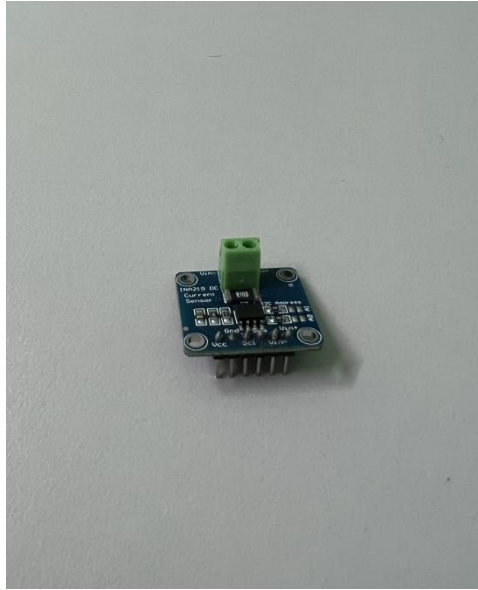


Figure 5.1.2.1: INA219 after soldering the small parts together

The hardware were assembled out based on the system design diagram illustrated in the previous chapter. The small components of the INA219 DC current sensor need to be soldered together as shown in Figure 5.1.2.1 to obtain a stable connection.



Figure 5.1.2.2: Type of precision screwdriver

The precision screwdriver as shown in Figure 5.1.2.2 is needed in order to connect the INA219 DC current sensor  $V_{in-}$  and  $V_{in+}$  ports to the breadboard and batteries respectively. A male-to-female jumper wire would be sufficient for the connection between the NodeMCU ESP32 and INA219 DC current sensor through the breadboard.

## 5.2 Coding

### 5.2.1 Arduino IDE

Coding is essential to obtain the Voltage, Current and Power flow readings from the batteries to the light bulb through the INA219 DC current sensor, as well as connecting the NodeMCU ESP32 board to the Blynk IoT Cloud too. The coding was written using Arduino IDE.

```
#define BLYNK_TEMPLATE_ID      "TMPLnAxZlQVr"
#define BLYNK_DEVICE_NAME     "Quickstart Template"
#define BLYNK_AUTH_TOKEN     "15Q6eCehDbAqLu0JFvnvsFnIBZOv4n9_"
```

Based on the code snippet above, the template ID, device name, and authorisation token were defined inside the code. The template ID and device name were obtained after creating a new template in the Blynk IoT web application. After creating a new account for the Blynk IoT, the authorisation token was obtained from the registered email's inbox.

```
#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <Wire.h>
#include <Adafruit_INA219.h>
```

The “BLYNK\_PRINT Serial” command is used to display the INA219 DC current sensor readings to the Serial Monitor. Few libraries were used, among are WiFi, WiFiClient and Wire libraries. BlynkSimpleEsp32 library is included, in order to connect and upload the code to the NodeMCU ESP32 board. Lastly, the Adafruit\_INA219 is included in order to use the pre-defined function to measure the Voltage, Current and Power readings.

```
Adafruit_INA219 ina219;
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "lee@unifi";
char pass[] = "onn7117663lee";
BlynkTimer timer;
```



Code snippet above shows the variables which were used in this project. The first variable: `ina219`, is used for pre-defined function. Meanwhile, the character type variables were used to access the Blynk IoT Cloud. Lastly, the timer variable was used to set a function to be called periodically.

```
void loop(){
  float shuntVoltage = 0;
  float loadVoltage = 0;
  float current_mA = 0;
  float batteryVoltage = 0;
  float power_mW = 0;

  Blynk.run();
  timer.run();

  SchedBase::dispatcher(); // dispatch any task due

  shuntVoltage = ina219.getShuntVoltage_mV(); // the voltage at shunt resistor
  terminals
  loadVoltage = ina219.getBusVoltage_V();
  current_mA = ina219.getCurrent_mA();
  power_mW = ina219.getPower_mW();
  batteryVoltage = loadVoltage + (shuntVoltage / 1000);

  Serial.print("Loads Voltage: "); Serial.print(loadVoltage); Serial.println(" V");
  Serial.print("Shunt Voltage: "); Serial.print(shuntVoltage); Serial.println(" mV");
  Serial.print("Battery Voltage: "); Serial.print(batteryVoltage); Serial.println("
  V");
  Serial.print("Current: "); Serial.print(current_mA); Serial.println(" mA");
  Serial.print("Power: "); Serial.print(power_mW); Serial.println(" mW");
  Serial.println("");
  delay(2000);
}
```

The code snippet above showed the pre-defined function that comes along from the example of the Adafruit INA219 library and also the SchedBase dispatcher function. A total of five variables were declared on the first few rows, which are the `shuntVoltage`, `loadVoltage`, `current_mA`, `batteryVoltage` and `power_mW`. The `shuntVoltage` represents the voltage at shunt resistor terminals. Next, the SchedBase dispatcher is used to dispatch any task due. Besides, the `loadVoltage` represents the voltage for the light bulb, which is the DC light bulb. Furthermore, the `current_mA` represent the current for the light bulb. In addition, the `power_mW` represent the

power generated for the circuit. Lastly, the batteryVoltage represent the voltage from the battery.

```
BLYNK_WRITE(V0) // Executes when the value of virtual
pin 0 changes{
  if(param.asInt() == 1){
    // execute this code if the switch widget is now ON
    digitalWrite(23,HIGH); // Set digital pin 23 HIGH
  }
  else{
    // execute this code if the switch widget is now OFF
    digitalWrite(23,LOW); // Set digital pin 23 LOW
  }
}
```

The first self-defined function is to toggle the on and off of the DC light bulb. The V0 pin from the previous step is used as a parameter. When the Virtual pin 0 is set to 1, it will set the NodeMCU ESP32 board's digital pin 23 to HIGH. It means that it will send a signal to the relay module and instruct the relay module to switch on the light bulb. Otherwise, digital pin 23 will be set to LOW. This means that the relay module will switch off the light bulb by default.

```
void readSensor(){
  float v = ina219.getBusVoltage_V();
  float c = ina219.getCurrent_mA();
  float p = ina219.getPower_mW();

  if (isnan(v) || isnan(c) || isnan(p)) {
    Serial.println("Failed to read from INA219 sensor");
    return;
  }

  Blynk.virtualWrite(V2, v);
  Blynk.virtualWrite(V3, c);
  Blynk.virtualWrite(V4, p);
}
```

The second self-defined function would be the `readSensor()`. The `readSensor()` is used to obtain the readings from the INA219 DC current sensor, and pass them to the Blynk IoT Cloud. The Virtual pin 2, 3, and 4 that were created in the previous chapter will represent different reading values, which are the voltage, current, and power. The `Blynk.virtualWrite()` is used to pass the data to the Blynk IoT Cloud virtual pin respectively.

```
void turnOnBulb (){
digitalWrite(23,ON);
}

void turnOffBulb (){
digitalWrite(23,OFF);
}

SchedTask OnTask (0, 3129, turnOnBulb);
SchedTask OffTask (1000, 3129, turnOffBulb);
```

Additionally, two self-defined functions were created for the scheduled task. These functions are also used to turn on and off the light bulb. After that, the functions will be defined as a parameter of the `SchedTask OnTask` and the `SchedTask OffTask`. The `SchedTask OnTask` is used to turn on the light bulb periodically every 3.129 seconds and vice versa on the `SchedTask OffTask`.

```
void energyConsumed(){
  kWh = kWh + ina219.getPower_mW()*(millis()-
lastmillis)/3600000000.0;
  Serial.print("Accumulated Energy: "); Serial.print(kWh, 4);
  Serial.println(" kWh");
  lastmillis = millis();
  Blynk.virtualWrite(V5, kWh);
}
```

The last self-defined function would be the `energyConsumed()` function. This function is created for accumulating the energy left of the battery after conducting the experiment. The accumulated energy was calculated in kWh.

```
void setup(){
  // Debug console
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass);

  uint32_t currentFrequency;

  Serial.println("Hello!");

  pinMode(23, OUTPUT); // bulb at D23

  ina219.begin();

  Serial.println("Measuring voltage and current with
  INA219 ...");

  timer.setInterval(1000L, myTimerEvent);
  timer.setInterval(2000L, readSensor);
}
```

From the code snippet above, the `Serial.begin(115200)` was declared to allow the message to be printed to the Serial Monitor with 115200 baud. Then, the code is followed by the `Blynk.begin(auth, ssid, pass)`. This is to pass the parameters such as authentication token, Wi-Fi SSID, and Wi-Fi password to the Blynk IoT Cloud console for connection.

The line with `pinMode(23, OUTPUT)` is used to set the digital pin 23 as an output pin, which is in a low-impedance state. This means they can supply a suitable amount of current to generate signals for the 4-channel relay module.

When the `ina219.begin()` function is called, the INA219 DC current sensor begins measuring the voltage and current. To avoid spamming the message to the serial monitor, `timer.setInterval(2000L, readSensor)` was declared.

### 5.2.2 PSO Visual Studio Code

The PSO algorithm was built inside the Visual Studio Code that is written in Python language. The PSO algorithm helped to generate a suitable time in seconds for the scheduler timer inside the Arduino IDE code.

```
def f(x,y):
    "Objective function"
    return -20.0 * exp(-0.2 * sqrt(0.5 * (x**2 + y**2)))-exp(0.5 *
(cos(2 * pi * x)+cos(2 * pi * y))) + e + 20

x_min = x.ravel()[z.argmin()] # minimization problem
y_min = y.ravel()[z.argmin()]
```

From the code snippet above, the objective function was defined in a computer-readable form. The objective function that was used in this project was the Ackley function. Next, the PSO algorithm are focusing on `min()` and `argmin()` functions here because the project objective was set as a minimisation problem. Moreover, the, `c1` and `c2` parameter was defined accordingly based on the suggestion by the previous researcher.

```
n_particles = 50
np.random.seed(100)
X = np.random.rand(2, n_particles) * 5
V = np.random.randn(2, n_particles) * 0.1
```

Besides, 50 particles were created at random locations in this region, together with random velocities sampled over a normal distribution with mean=5 and standard deviation=0.1.

```

pbest = X
pbest_obj = f(X[0], X[1])
gbest = pbest[:, pbest_obj.argmin()]
gbest_obj = pbest_obj.min()

```

The vector `pbest_obj` is the best value found by each particle of the objective function. Similarly, `gbest_obj` is the best scalar value ever found by the swarms of the objective function.

```

def update():
    "Function to do one iteration of particle swarm optimization"
    global V, X, pbest, pbest_obj, gbest, gbest_obj
    # Update parameters
    r1, r2 = np.random.rand(2)
    V = w * V + c1*r1*(pbest - X) + c2*r2*(gbest.reshape(-1,1)-X)
    X = X + V
    obj = f(X[0], X[1])
    pbest[:, (pbest_obj >= obj)] = X[:, (pbest_obj >= obj)]
    pbest_obj = np.array([pbest_obj, obj]).min(axis=0)
    gbest = pbest[:, pbest_obj.argmin()]
    gbest_obj = pbest_obj.min()

    print("Most optimum timer value at f({})={}".format(gbest,
    gbest_obj))

```

The `update()` function above was used to assist the iteration of the PSO algorithm until all the particles had converged into a single point, which is the global minimum. First of all, all the needed variables were defined such as `V`, `X`, `pbest`, `pbest_obj`, `gbest`, and `gbest_obj`. Based on what had mentioned in Chapter 2.3, the particle velocity needs to be updated by using Formula (1) and the particle position

needs to be updated by using Formula (2). Thus, the velocity will be stored inside  $V$  and the particle position will be stored inside  $X$ .

In each iteration, after all the particles had searched through the region, they would need to update their personal best velocity and position, and the values should be stored inside  $pbest\_obj$ . If found that certain particle had the best velocity or best position explored, the values need to be stored inside  $gbest\_obj$ . Lastly, the last line was used to print out the most optimum timer value for the objective function.

### 5.3 System Operation

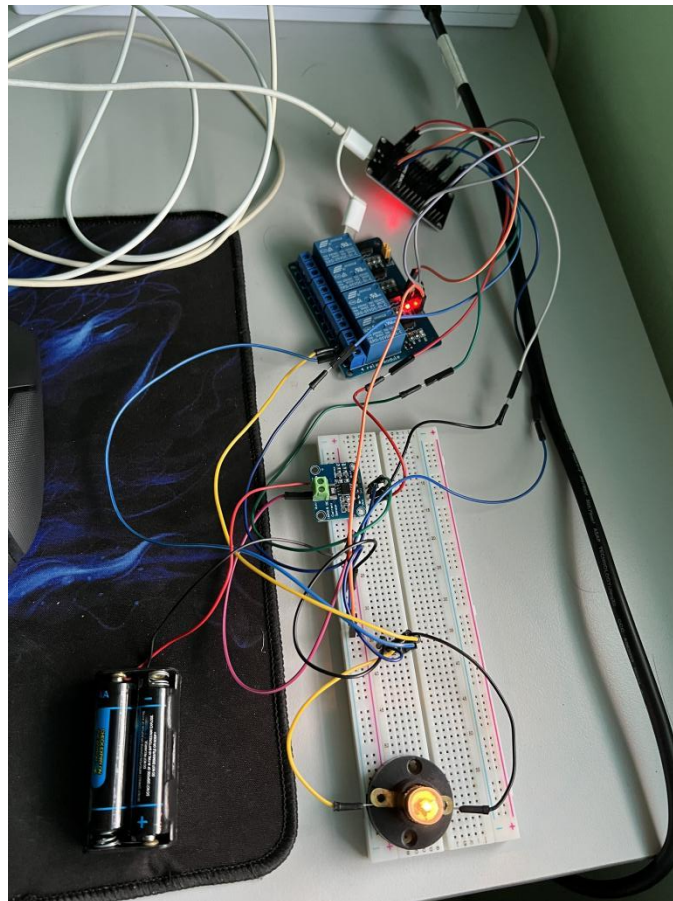


Figure 5.3.1: System circuit for operation

This sub chapter discussed the result achieved of the pre-development. This project will mainly focus on record the light bulb's voltage, current and power. The mobile application prototype will display these three readings in real time.

Figure 5.3.1 above showed the system circuit was assemble based on the system architecture diagram in Chapter 3.

Since the data acquired were highly reliable, therefore the calibration of the INA219 DC current sensor was required.

### When light bulb is switched off:

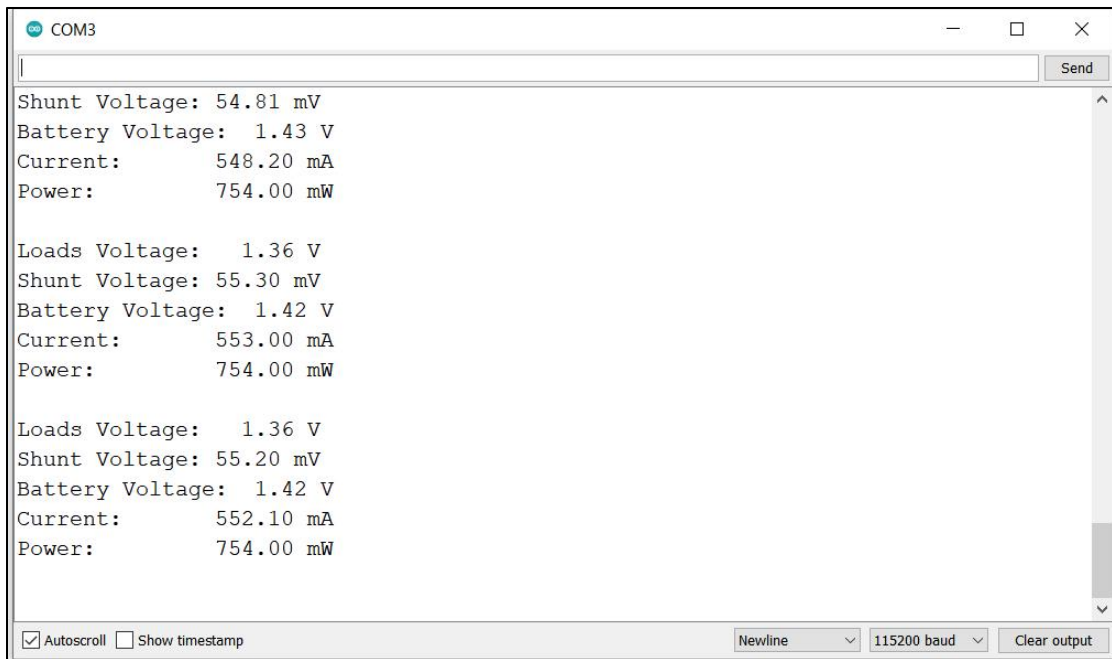


Figure 5.3.2: Results obtained from Serial Monitor in switched off condition



Tseu Kwan Lee

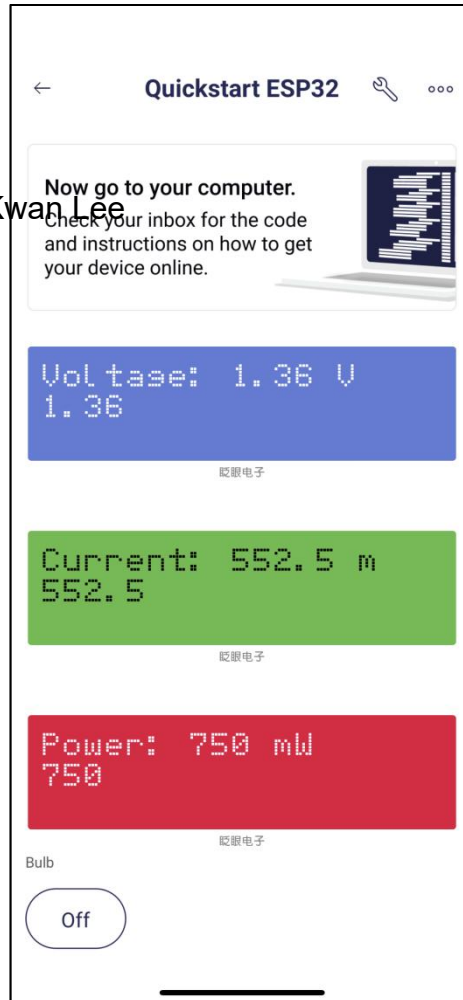


Figure 5.3.3: Results obtained from the mobile application prototype in switched off condition

When the light bulb is switched off, the readings begin to appear in real time. Since the batteries were connected to the circuit, some current still will flow through the circuit even when the light bulb is switched off.

**When light bulb is switched on:**

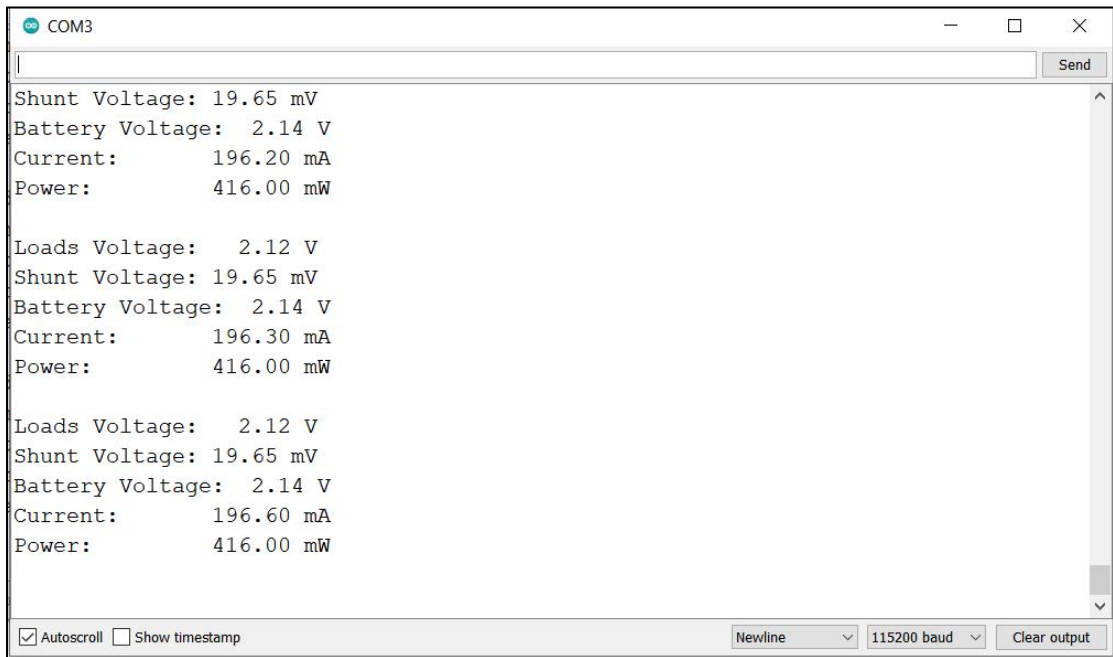


Figure 5.3.4: Results obtained from Serial Monitor in switched on condition

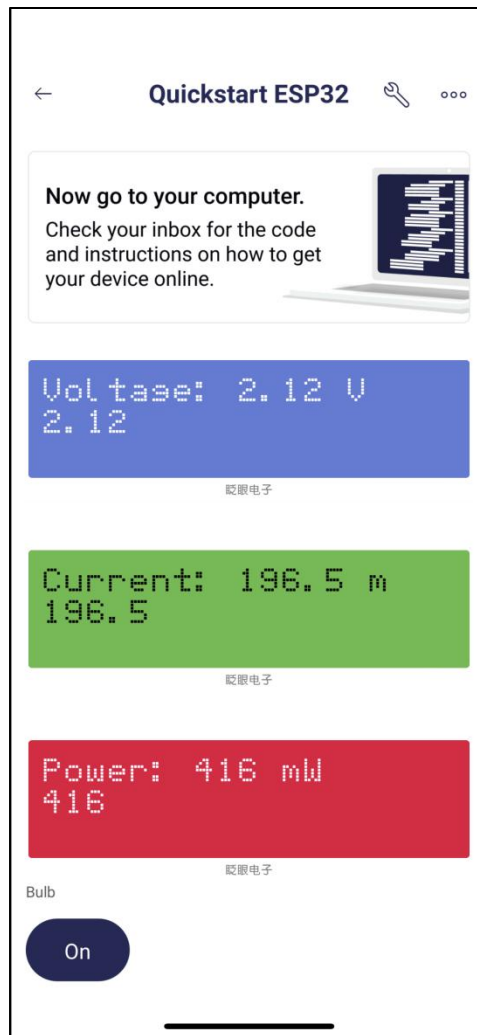


Figure 5.3.5: Results obtained from the mobile application prototype in switched on condition

When the light bulb is switched on, the voltage begin to rise. Based on the Ohm law equation [29], the current should be lowered as well. Hence, the power should be reduced too.

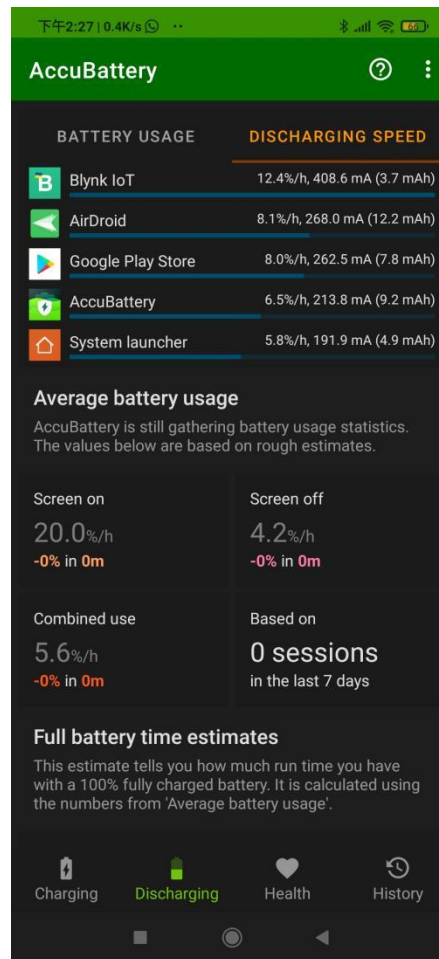


Figure 5.3.6: Results obtained from the AccuBattery

Figure 5.3.6 above depicts the result collected from the AccuBattery energy profiler application. The application itself will monitor the discharge speed and quantity for each of the application that were installed in the mobile phone. For example, the Blynk IoT program has drain up 3.7mAh from the total of 3300mAh of the Android smart phone battery life.

## **CHAPTER 6**

### **System Evaluation And Discussion**

#### **6.1 System Testing and Performance Metrics**

In this project, the testing method would be performed by comparing the EC in two situations, which are with and without applying the SI-based scheduling task to the circuit. The testing period for each situation is allocated for five minutes. This is because the circuit was run with DC batteries, hence the long period would not be suitable to prevent the batteries voltage to drained out. A timer would be used for both experiment conditions in order to obtain a fair and accurate result.

The performance of each experiment condition would be evaluated by observing the “Accumulated Energy” value. The “Accumulated Energy” value is representing the energy left inside the batteries after five minutes. The value is converted into kWh for easy calculation.

## 6.2 Testing Setup and Result

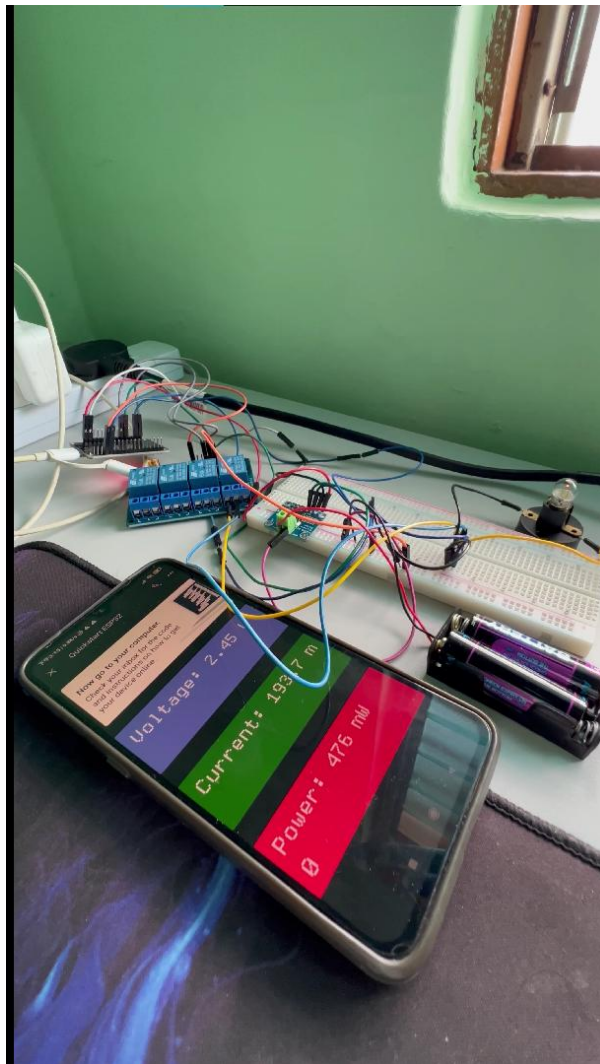


Figure 6.2.1: Testing environment setup

Figure 6.2.1 above shows the testing environment setup when performing the testing. The comparison had been done before and after applying the SI-based scheduling task to the circuit. After applying the SI-based scheduling task to the circuit was defined as Experiment 1, and before applying the SI-based scheduling task to the circuit was defined as Experiment 2.

## CHAPTER 6

### Experiment 1: SI scheduling task

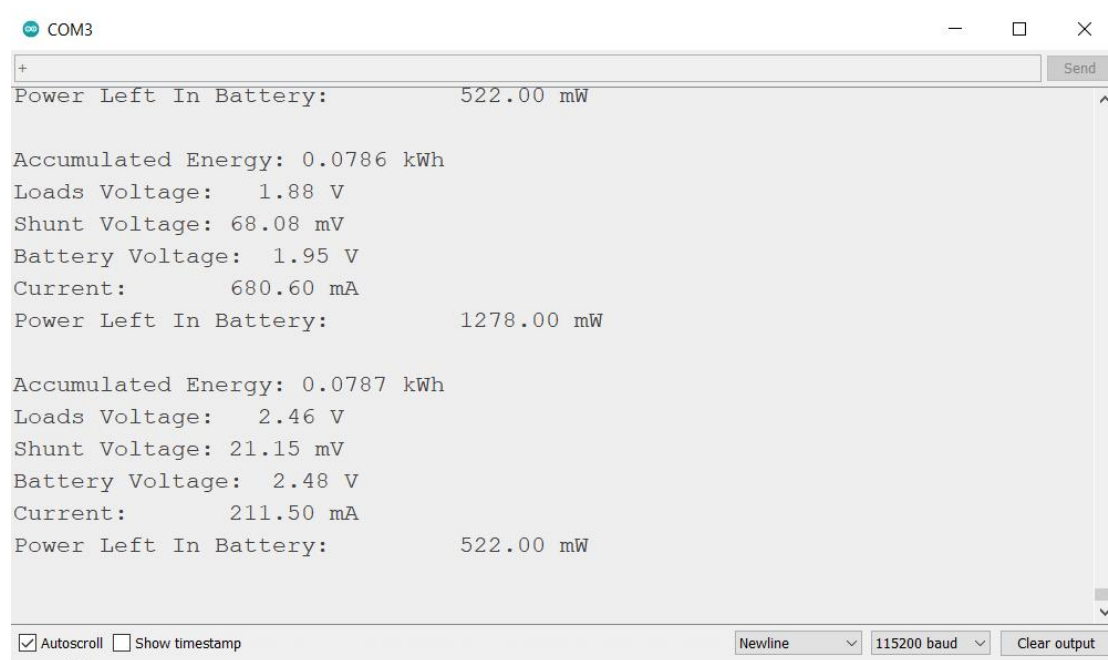


Figure 6.2.2: Experiment 1 final result in five minutes period

### Experiment 2: Non SI scheduling task

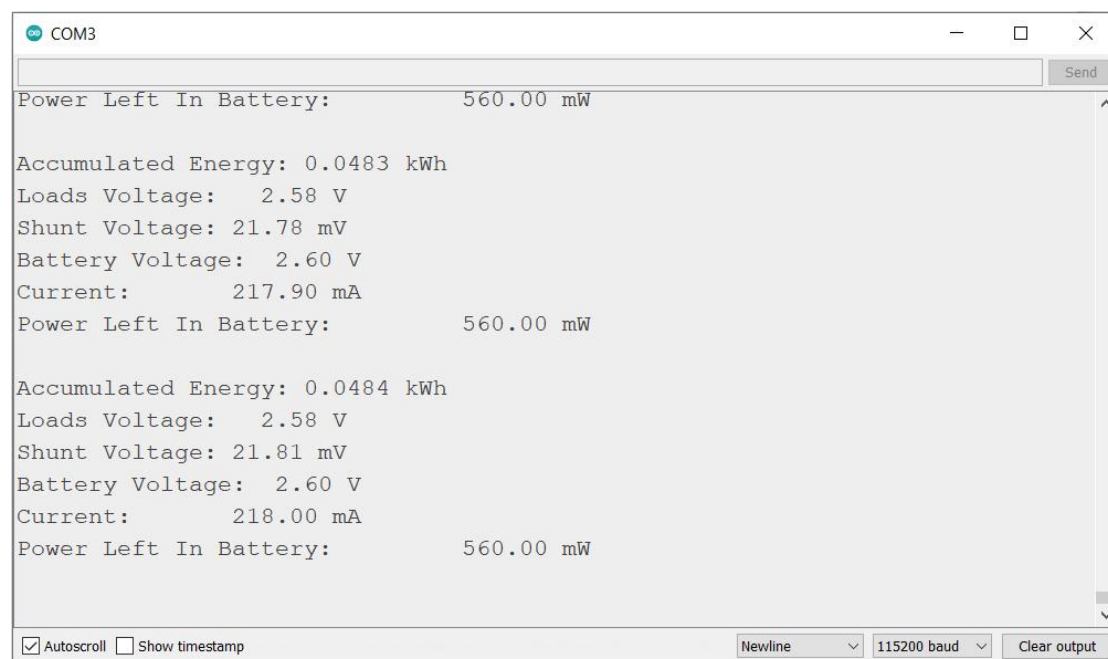


Figure 6.2.3: Experiment 2 final result in five minutes period

## CHAPTER 6

Notice that the “Accumulated Energy” value in the Experiment 1 was higher than in the Experiment 2. Below was a simple calculation on how much percentage of energy was saved in the testing result obtained.

Energy calculation in percentage:  $0.0484/0.0787*100\% = \text{save } 61.34\% \text{ of energy}$

Period of time	Accumulated energy(kWh)
1 minute	0.0186
2 minutes	0.0293
3 minutes	0.0426
4 minutes	0.0577
5 minutes	0.0787

Table 6.2.1: Experiment 1 results in different period of time

Period of time	Accumulated energy(kWh)
1 minute	0.0088
2 minutes	0.0170
3 minutes	0.0249
4 minutes	0.0329
5 minutes	0.0484

Table 6.2.2: Experiment 2 results in different period of time



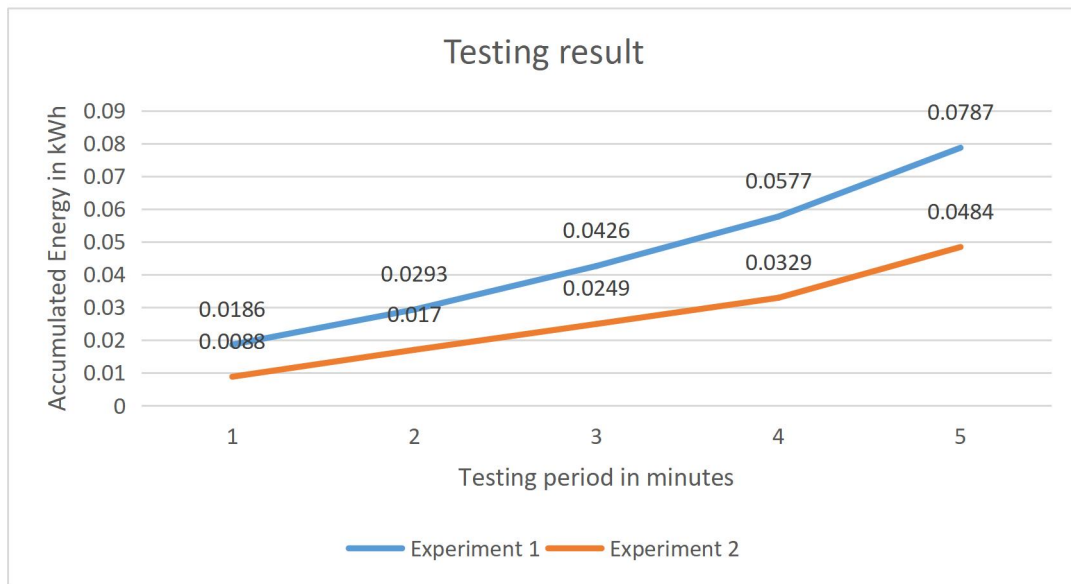


Figure 6.2.4: Comparison of both experiment conditions testing result

In conclusion, the SI-based scheduling task was proved able to optimize the EC of the light bulb in five minutes period.



Figure 6.2.5: AccuBattery application result in Experiment 1

The Android smartphone battery drained when performing the Experiment 1 testing was recorded as 34.9mAh, in total of 3300mAh.



Figure 6.2.6: AccuBattery application result in Experiment 2

The Android smartphone battery drained when performing the Experiment 2 testing was recorded as  $(62.6-34.9) = 27.7\text{mAh}$ , in total of 3300mAh.

Based on the results in Figure 6.2.5 and Figure 6.2.6, the EC in experiment 1 was slightly higher than the EC in Experiment 2. This project concluded that the EC of mobile application was affected by the SI-based scheduler task.

### 6.3 Project Challenges

There are several obstacles experienced while implementing this project. The first problem would be the difficulties in procuring the IoT hardware. This is due to a lack of IoT hardware, such as the NodeMCU ESP32 board, INA219 DC current sensor, light bulb, battery holder, and so on, in the UTAR FICT lab. Obtaining these IoT devices is time-consuming as most of them need to be acquired through an online buying platform. Among the troubles encountered when purchasing hardware items via an online shopping platform are delays in package delivery, partial parcel received, and faulty items received.

The second problem would be deciding which instruments to use to monitor EC. In terms of the IoT hardware side, the choice of selecting the IoT hardware for the implementation tools is significant since it may save a lot of time. For example, the choice of the light bulb whether is an AC type or a DC type electrical appliances need to consider thoughtfully since the sensors for monitoring the light bulb are different. In terms of the software side, which is the energy profiler. Many of the energy profilers that use to monitor EC are mainly monitoring on the web application side, rather than the mobile application development side. As a consequence, learning the correct instrument was time intensive too.

When experimenting, there are a few challenges that would be faced. The first challenge would be unstable results collection. This is because the circuit is built using DC. As the experiment keeps on going, the battery will start to be drained out. Hence, the accumulated energy obtained after a certain time will be not accurate, and it may affect the calculation of the energy saved before and after applying the SI-based scheduling task.

Besides, another challenge that would be faced in this project is the delay of the microprocessor. Since this project is using a relay module to switch on and off the light bulb, the signal should be sent out to the relay module from the microprocessor consistently based on the scheduled task timer. However, there is some delay that happened between the microprocessor and relay module and it may affect the consistency of the system to record the “Accumulated Energy”.

Lastly, this project would be using Python code to implement the PSO algorithm. Since the code built previously is mainly focused on Arduino IDE, which is using the C++ programming language. Hence, some interoperability issues will occur when both of the codes run together. This is due to the reason that the NodeMCU is unable to read the flash memory when trying to implement the PSO code inside the Arduino IDE. The possible reason would be a power issue. If the board has other hardware connected, they may be drawing too much power from the rail as they start up.

### **6.4 Objectives Evaluation**

The first objective is to study and understand the existing sources of applying the swarm intelligence algorithm in the Smart Home scheduling task application. This objective had been achieved by finding the resources such as research papers, online materials, and so on to construct the literature review.

The second objective is to propose a Swarm Intelligence-based scheduling task to optimise the Smart Home appliance's EC. This had been achieved by developing the scheduling task inside the Arduino IDE. To let the scheduling task get involved with Swarm Intelligence, this project separately generated a suitable timing in Visual Studio Code for the scheduling task with the help of the PSO algorithm.

The third objective is to develop a Swarm Intelligence-based scheduling task mobile application prototype for Smart Home users. As shown in Chapter 5.3, the Swarm Intelligence-based scheduling task mobile application prototype had been created to monitor the voltage and current value and allow user to switch on-and-off the light bulb.

The last objective is to evaluate the performance of the proposed method in optimising EC. This was also achieved in Chapter 6.2 by observing the “Accumulated Energy” value in both Experiment 1 and Experiment 2.

### **6.5 Concluding Remark**

Based on the objective evaluation above, some concluding remarks could be said for this project. First of all, the objectives were fulfilled based on the first image of the project owner, without amending them due to personal constraints. However,

## CHAPTER 6

since the testing metric was purely built by the project owner himself, hence it may not much accurate and convincing because the method used doesn't follow any international standard.

## CHAPTER 7

### 7.1 Conclusion

When talking about IoT devices, people may think of IoT devices such as sensors, actuators, gadgets, appliances, or machines as pieces of hardware, that are configured for specific applications and can transfer data via the internet or other networks. The idea behind the IoT is to have a device that can self-learn in real-time, enhancing efficiency and bringing crucial information to the surface faster than a system reliant on human interaction. However, individuals may only think of the benefits offered by the IoT devices, but they seldom ponder about the EC of the IoT devices. Just imagine, if every resident implements the smart home design on their property, how much energy will it drain up, and how much harm will be done to the environment.

Thus, this project was presented to help the Smart Home user in improving the EC of these IoT devices in their Smart Home. The proposed solution would be utilising the Swarm Intelligence algorithm to establish a task schedule. This strategy may alleviate consumer annoyance in optimising the EC since some of them might or do not have a solid concept on how to achieve it.

In this report, the first goal is to imitate a Smart Home environment by designing the system with smart home IoT hardware. The light bulb from the circuit should be able to toggle on and off. The second purpose is to record the voltage and current readings of the circuit's light bulb. The third purpose is to design a mobile application prototype to control the IoT device and display the device statistics data such as the voltage, current, and power.

The results were obtained by observing the voltage, current and power readings from the mobile application prototype. The readings could be obtained after connecting the USB cable to the NodeMCU ESP32 board and attaching the batteries to power up the light bulb. The readings obtained were quite stable when toggling on and off of the light bulb.

Following that, the additional coding section would be developing the SI-based scheduling task. Hence, the PSO algorithm was studied and successfully

incorporate it into the task schedule. After constructing the SI-based task schedule, the comparison between the EC when applying the SI-based task schedule and without applying the SI-based task schedule to the light bulb was recorded.

In conclusion, based on the testing results obtained, this project had proven that the SI-based scheduler task is able to help in optimising the EC of the Smart Home appliances.

### **7.2 Recommendation**

The first recommendation for future work would be swapping the whole DC circuit for the AC circuit. Due to the budget and safety issues, this project had to simulate the Smart Home environment using the DC circuit. However, if following the standard of the Smart Home environment, the electrical appliances should be mainly built based on the AC power supply mode. This is to let the electrical appliances have stable electric flow and supply. The DC battery inside those electrical appliances is most likely used as a short-term backup during a blackout. In short, the AC circuit would help to generate a more stable result during the experiment being conducted.

The second recommendation for future work would be to implement the project using a consistent programming language. This action would help to let the hardware, mobile application, and scheduling task to achieve a stable interoperability. Hence, the system should be able to conduct dynamically without any manual input for saving some time.

The last recommendation for future work would be the SI algorithm. Due to the time constraint, this project only manages to build a single PSO algorithm and apply it to the scheduler task. In the future, some other SI algorithms such as the ACO algorithm can be implemented to make a comparison between the performance of the PSO and the ACO when applying them to the scheduler task.



## REFERENCES

## REFERENCES

- [1] Hassan, JS, Zin, RM, Abd Majid, MZ, Balubaid, S & Hainin, “Building Energy Consumption in Malaysia: An Overview”, *Jurnal Teknologi*, vol. 70, no. 7, Oct. 2014. Accessed on: April 3, 2022. [Online]. Available doi: 10.11113/jt.v70.3574
- [2] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013, doi: 10.1016/j.future.2013.01.010.
- [3] “NP-complete problem | mathematics,” *Encyclopedia Britannica*. Accessed: April 3, 2022. [Online]. Available: <https://www.britannica.com/science/NP-complete-problem>
- [4] S. N. Makhadmeh, A. T. Khader, M. A. Al-Betar, and S. Naim, “Multi-objective power scheduling problem in smart homes using grey wolf optimiser,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 9, pp. 3643–3667, Oct. 2018, doi: 10.1007/s12652-018-1085-8
- [5] W. Sun, M. Tang, L. Zhang, Z. Huo, and L. Shu, “A Survey of Using Swarm Intelligence Algorithms in IoT,” *Sensors*, vol. 20, no. 5, p. 1420, Mar. 2020, doi: 10.3390/s20051420.
- [6] Y.-H. Lin and Y.-C. Hu, “Residential Consumer-Centric Demand-Side Management Based on Energy Disaggregation-Piloting Constrained Swarm Intelligence: Towards Edge Computing,” *Sensors*, vol. 18, no. 5, p. 1365, Apr. 2018, doi: 10.3390/s18051365.
- [7] D. Mahmood *et al.*, “Realistic Scheduling Mechanism for Smart Homes,” *Energies*, vol. 9, no. 3, p. 202, Mar. 2016, doi: 10.3390/en9030202.
- [8] “Energy Malaysia Towards a World-Class Energy Sector Towards a Brighter Future Energy in Malaysia.” [Online]. Available:

## REFERENCES

- <https://www.st.gov.my/en/contents/publications/energyMalaysia/EM12%20Nov%202017%20v2.pdf>
- [9] C. Koliás, G. Kambourakis, and M. Maragoudakis, “Swarm intelligence in intrusion detection: A survey,” *Computers & Security*, vol. 30, no. 8, pp. 625–642, Nov. 2011, doi: 10.1016/j.cose.2011.08.009.
- [10] A. Colomi, M. Dorigo and V. Maniezzo, “Distributed optimization by ant colonies,” in *Proc. of ECAL-91 - European Conference On Artificial Life, Paris, France, Elsevier Publishing*, Jan, 1991. pp.134–142 [Online]. Available: [https://www.researchgate.net/publication/216300484\\_Distributed\\_Optimization\\_by\\_Ant\\_Colonies](https://www.researchgate.net/publication/216300484_Distributed_Optimization_by_Ant_Colonies)
- [11] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proc. of ICNN'95 - International Conference on Neural Networks*, 1995, pp. 1942-1948 vol.4, doi: 10.1109/ICNN.1995.488968.
- [12] K. M. Passino, “Bacterial Foraging Optimization,” *Innovations and Developments of Swarm Intelligence Applications*, pp. 219–234, doi: 10.4018/978-1-4666-1592-2.ch013.
- [13] Xiao Lei LI;Zhi Jiang SHAO;Ji Xin QIAN. An Optimizing Method Based on Autonomous Animats: Fish-swarm Algorithm. *Systems Engineering - Theory & Practice*. 2002, 22(11): 32-38 [https://doi.org/10.12011/1000-6788\(2002\)11-32](https://doi.org/10.12011/1000-6788(2002)11-32)
- [14] D. Karaboga and B. Basturk, “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm,” *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, Apr. 2007, doi: 10.1007/s10898-007-9149-x.
- [15] X.-S. Yang, “Nature-Inspired Metaheuristic Algorithms Second Edition.” [Online]. Available: [https://staff.fmi.uvt.ro/~daniela.zaharie/ma2016/projects/techniques/FireflyAlgorithm/Yang\\_nature\\_book\\_part.pdf](https://staff.fmi.uvt.ro/~daniela.zaharie/ma2016/projects/techniques/FireflyAlgorithm/Yang_nature_book_part.pdf)
- [16] H. Duan and P. Qiao, “Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning,” *International Journal*

## REFERENCES

- of Intelligent Computing and Cybernetics*, vol. 7, no. 1, pp. 24–37, Mar. 2014, doi: 10.1108/ijicc-02-2014-0005.
- [17] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey Wolf Optimizer,” *Advances in Engineering Software*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.
- [18] Arora, S & Singh, S, “Butterfly optimization algorithm: a novel approach for global optimization” *Journal of Soft Computing*, vol. 23, no. 3, pp. 715–734, March 2018. Accessed on: April. 3, 2022. [Online]. Available <https://doi.org/10.1007/s00500-018-3102-4>
- [19] S. N. Makhadmeh, A. T. Khader, M. A. Al-Betar, S. Naim, A. K. Abasi, and Z. A. A. Alyasseri, “Optimization methods for power scheduling problems in smart home: Survey,” *Renewable and Sustainable Energy Reviews*, vol. 115, p. 109362, Nov. 2019, doi: 10.1016/j.rser.2019.109362.
- [20] N. Javaid *et al.*, “A Hybrid Genetic Wind Driven Heuristic Optimization Algorithm for Demand Side Management in Smart Grid,” *Energies*, vol. 10, no. 3, p. 319, Mar. 2017, doi: 10.3390/en10030319.
- [21] M. H. Rahim, A. Khalid, N. Javaid, M. Alhussein, K. Aurangzeb, and Z. A. Khan, “Energy Efficient Smart Buildings Using Coordination Among Appliances Generating Large Data,” *IEEE Access*, vol. 6, pp. 34670–34690, 2018, doi: 10.1109/ACCESS.2018.2805849.
- [22] S. Rahim, N. Javid, A. Ahmad, S. Ahmed Khan, Z. Ali Khan, N. Alrajeh, U. Qasim, “Exploiting heuristic algorithms to efficiently utilize energy management controllers with renewable energy sources,” *Energy and Buildings*, vol. 129, pp. 452–470, Oct. 2016, doi: 10.1016/j.enbuild.2016.08.008.
- [23] M. Tariq, A. Khalid, I. Ahmad, M. Khan, B. Zaheer & N. Javaid, “Load Scheduling in Home Energy Management System Using Meta-Heuristic Techniques and Critical Peak Pricing Tariff,” in *Proc. 12th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC-2017)*, Spain, pp. 50–62. [Online]. Available:

## REFERENCES

- [https://www.researchgate.net/publication/319331395\\_Load\\_scheduling\\_in\\_home\\_energy\\_management\\_system\\_using\\_meta-heuristic\\_techniques\\_and\\_critical\\_peak\\_pricing\\_tariff](https://www.researchgate.net/publication/319331395_Load_scheduling_in_home_energy_management_system_using_meta-heuristic_techniques_and_critical_peak_pricing_tariff)
- [24] M. A. Khan, N. Javaid, A. Mahmood, Z. A. Khan, and N. Alrajeh, "A generic demand-side management model for smart grid," *International Journal of Energy Research*, vol. 39, no. 7, pp. 954–964, Mar. 2015, doi: 10.1002/er.3304.
- [25] M. Pedrasa, T. D. Spooner, and I. F. MacGill, "Scheduling of Demand Side Resources Using Binary Particle Swarm Optimization," *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1173–1181, Aug. 2009, doi: 10.1109/tpwrs.2009.2021219.
- [26] Y. Zhou, Y. Chen, G. Xu, Q. Zhang, and L. Krundel, "Home energy management with PSO in smart grid," *IEEE Xplore*, Jun. 01, 2014. <https://ieeexplore.ieee.org/document/6864865> (accessed Aug. 10, 2021).
- [27] N. Bergmann *et al.*, "Using Swarm Intelligence to Optimize the Energy Consumption for Distributed Systems," *Modern Applied Science*, vol. 7, no. 6, May 2013, doi: 10.5539/mas.v7n6p59.
- [28] T. Maharani, M. Abdurohman, and A. G. Putrada, "Smart Lighting in Corridor Using Particle Swarm Optimization," *2019 Fourth International Conference on Informatics and Computing (ICIC)*, Oct. 2019, doi: 10.1109/icic47613.2019.8985858.
- [29] All About Circuits, "Ohm's Law - How Voltage, Current, and Resistance Relate | Ohm's Law | Electronics Textbook," *Allaboutcircuits.com*, 2018. <https://www.allaboutcircuits.com/textbook/direct-current/chpt-2/voltage-current-resistance-relate/>
- [30] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, 2019, doi:

## REFERENCES

10.1109/icec.1998.699146.

- [31] S. Surjanovic and D. Bingham, "Ackley Function," *Simon Fraser University*, 2013. <https://www.sfu.ca/~ssurjano/ackley.html> (accessed Aug. 24, 2022).

**APPENDIX**

**FINAL YEAR PROJECT WEEKLY REPORT 1**

*(Project I)*

<b>Trimester, Year: 2, 3</b>	<b>Study week no.: 1-4</b>
<b>Student Name &amp; ID: Lee Kah Poh 19ACB06226</b>	
<b>Supervisor: Ms. Tseu Kwan Lee</b>	
<b>Project Title: USING SWARM INTELLIGENCE TECHNIQUE TO OPTIMIZE THE ENERGY CONSUMPTION OF SMART HOME APPLICATION ARCHITECTURE</b>	

**1. WORK DONE**

[Please write the details of the work done in the last fortnight.]

- Studied on research paper about the PSO algorithm
- Studied online materials on how to build the scheduling task
- Draw out the related diagram needed


**2. WORK TO BE DONE**

Develop the scheduling task

**3. PROBLEMS ENCOUNTERED**

**4. SELF EVALUATION OF THE PROGRESS**

- 25% of overall progress



Supervisor's signature



Student's signature

## FINAL YEAR PROJECT WEEKLY REPORT 2

*(Project I)*

<b>Trimester, Year: 2, 3</b>	<b>Study week no.: 4-6</b>
<b>Student Name &amp; ID: Lee Kah Poh 19ACB06226</b>	
<b>Supervisor: Ms. Tseu Kwan Lee</b>	
<b>Project Title: USING SWARM INTELLIGENCE TECHNIQUE TO OPTIMIZE THE ENERGY CONSUMPTION OF SMART HOME APPLICATION ARCHITECTURE</b>	

### 1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Developed the scheduling task that can manually insert the timer
- Developed the function to record the energy remain left inside the battery in kWh.

### 2. WORK TO BE DONE

- Develop the PSO code for this project

### 3. PROBLEMS ENCOUNTERED

- Finding the solution to make the PSO code have interoperability issue with the Arduino IDE.

### 4. SELF EVALUATION OF THE PROGRESS

- 50% of overall progress



Supervisor's signature



Student's signature

## FINAL YEAR PROJECT WEEKLY REPORT 3

*(Project I)*

<b>Trimester, Year: 2, 3</b>	<b>Study week no.: 6-8</b>
<b>Student Name &amp; ID: Lee Kah Poh 19ACB06226</b>	
<b>Supervisor: Ms. Tseu Kwan Lee</b>	
<b>Project Title: USING SWARM INTELLIGENCE TECHNIQUE TO OPTIMIZE THE ENERGY CONSUMPTION OF SMART HOME APPLICATION ARCHITECTURE</b>	

### 1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Developed the PSO code in Python programming language

### 2. WORK TO BE DONE

- Run the testing in both experiment condition, which is with and without applying the SI-based scheduling task into the circuit

### 3. PROBLEMS ENCOUNTERED

- NA

### 4. SELF EVALUATION OF THE PROGRESS

- 70% of overall progress



Supervisor's signature



Student's signature



**FINAL YEAR PROJECT WEEKLY REPORT 4***(Project I)*

<b>Trimester, Year: 2, 3</b>	<b>Study week no.: 8-10</b>
<b>Student Name &amp; ID: Lee Kah Poh 19ACB06226</b>	
<b>Supervisor: Ms. Tseu Kwan Lee</b>	
<b>Project Title: USING SWARM INTELLIGENCE TECHNIQUE TO OPTIMIZE THE ENERGY CONSUMPTION OF SMART HOME APPLICATION ARCHITECTURE</b>	

**1. WORK DONE**

[Please write the details of the work done in the last fortnight.]

- Finished writing all the testing results and relevant details into the report

**2. WORK TO BE DONE**

- Submit to supervisor for checking

**3. PROBLEMS ENCOUNTERED**


- NA

**4. SELF EVALUATION OF THE PROGRESS**

- 100% of overall progress

\_\_\_\_\_  
Supervisor's signature\_\_\_\_\_  
Student's signature

POSTER




## USING SWARM INTELLIGENCE TECHNIQUE TO OPTIMIZE THE ENERGY CONSUMPTION OF SMART HOME APPLICATION ARCHITECTURE

### 1. Introduction

Individuals may only think about the benefits offered by the IoT devices, but they seldom ponder about the energy consumption between the IoT devices

### 2. Objectives

To develop a Swarm Intelligence-based scheduling task mobile application prototype for Smart Home user



**Assemble IoT hardware**


**Create mobile app**

**Link IoT hardware to Cloud**

**Refine result obtain**

**Allow user control appliances**

**User obtain energy result**



### 3. Conclusion

The objectives were achieved by observing the readings from the mobile application prototype and the testing results obtained

**Project Developer: Lee Kah Poh**  
**Project Supervisor: Ms. Tseu Kwan Lee**

## PLAGIARISM CHECK RESULT

## PLAGIARISM CHECK RESULT

ORIGINALITY REPORT			
8%	Tse	5%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS
PRIMARY SOURCES			
1	<a href="https://machinelearningmastery.com">machinelearningmastery.com</a> Internet Source	2%	
2	<a href="https://eprints.utar.edu.my">eprints.utar.edu.my</a> Internet Source	1%	
3	Danish Mahmood, Nadeem Javaid, Nabil Alrajeh, Zahoor Khan, Umar Qasim, Imran Ahmed, Manzoor Ilahi. "Realistic Scheduling Mechanism for Smart Homes", Energies, 2016 Publication	1%	
4	Sharif Naser Makhadmeh, Ahamad Tajudin Khader, Mohammed Azmi Al-Betar, Syibrah Naim et al. "Optimization methods for power scheduling problems in smart home: Survey", Renewable and Sustainable Energy Reviews, 2019 Publication	<1%	
5	<a href="https://miliohm.com">miliohm.com</a> Internet Source	<1%	
6	<a href="https://docs.blynk.io">docs.blynk.io</a> Internet Source	<1%	
7	<a href="https://www.coursehero.com">www.coursehero.com</a> Internet Source	<1%	

## PLAGIARISM CHECK RESULT

8	Harshavardham Reddy, Neerav Negi, Zeenia Gupta, Sagar Sood, Ishika Kansal, Nikhil Aggarwal. "Advanced IOT Home Automation Using Google Assistant and ThingSpeak IOT Platform", 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), 2022 Publication	<1%
9	github.com Internet Source	<1%
10	"Innovations in Smart Cities Applications Edition 3", Springer Science and Business Media LLC, 2020 Publication	<1%
11	Murad Khan, Bhagya Nathali Silva, Kijun Han. "Internet of Things Based Energy Aware Smart Home Control System", IEEE Access, 2016 Publication	<1%

## PLAGIARISM CHECK RESULT


---

12	<a href="https://community.createlabz.com">community.createlabz.com</a> Internet Source	<1 %
13	Bergmann, Neil, Yuk Ying Chung, Xiangrui Yang, Zhe Chen, Wei Chang Yeh, Xiangjian He, and Raja Jurdak. "Using Swarm Intelligence to Optimize the Energy Consumption for Distributed Systems", Modern Applied Science, 2013. Publication	<1 %
14	<a href="https://forum.arduino.cc">forum.arduino.cc</a> Internet Source	<1 %
15	Ghulam Hafeez, Noor Islam, Ammar Ali, Salman Ahmad, Muhammad Usman and Khurram Saleem Alimgeer. "A Modular Framework for Optimal Load Scheduling under Price-Based Demand Response Scheme	<1 %

---



## PLAGIARISM CHECK RESULT

	 Salman Ahmad, Muhammad Usman and Khurram Saleem Alimgeer. "A Modular Framework for Optimal Load Scheduling under Price-Based Demand Response Scheme in Smart Grid", Processes, 2019 Publication	
16	"Intelligent Computing Theories and Technology", Springer Nature, 2013 Publication	<1 %
17	<a href="http://fict.utar.edu.my">fict.utar.edu.my</a> Internet Source	<1 %
18	<a href="http://www.citizendia.org">www.citizendia.org</a> Internet Source	<1 %
19	Guido Cavraro, Andrey Bernstein, Vassilis Kekatos, Yingchen Zhang. "Real-Time Identifiability of Power Distribution Network Topologies With Limited Monitoring", IEEE Control Systems Letters, 2020 Publication	<1 %

## PLAGIARISM CHECK RESULT

20	I. Abd Latiff, M. O. Tokhi. "Fast convergence strategy for Particle Swarm Optimization using spread factor", 2009 IEEE Congress on Evolutionary Computation, 2009 Publication	<1%
21	<a href="http://www.anakkendali.com">www.anakkendali.com</a> Internet Source	<1%
22	James Shelley, Hawzhin Mohammed, Lizzy Zink, Syed Rafay Hasan, Omar Elkeelany. "Covert Communication Channel Detection in Low-Power Battery Operated IoT Devices: Leveraging Power Profiles", SoutheastCon 2018, 2018 Publication	<1%
23	<a href="http://components101.com">components101.com</a> Internet Source	<1%

## PLAGIARISM CHECK RESULT

24	gadgetshop.co.ke Internet Source	<1 %
25	"Intelligent and Fuzzy Techniques for Emerging Conditions and Digital Transformation", Springer Science and Business Media LLC, 2022 Publication	<1 %
26	Shaban R. S. Aldhshan, Khairul Nizam Abdul Maulud, Wan Shafrina Wan Mohd Jaafar, Othman A. Karim, Biswajeet Pradhan. "Energy Consumption and Spatial Assessment of Renewable Energy Penetration and Building Energy Efficiency in Malaysia: A Review", Sustainability, 2021 Publication	<1 %



## PLAGIARISM CHECK RESULT

---

27	<a href="http://webthesis.biblio.polito.it">webthesis.biblio.polito.it</a> Internet Source	<1 %
28	"Book of abstracts of 50th IEEE conference on decision and control and European control conference", IEEE Conference on Decision and Control and European Control Conference, 12/2011 Publication	<1 %
29	Salvatore, Dominick. "Managerial Economics in a Global Economy", Oxford University Press Publication	<1 %
30	Basar Koc, Dilip Sarkar, Huseyin Kocak, Ziya Arnavut. "A study of power consumption on MSP432 family of microcontrollers for lossless data compression", 2015 12th International Conference on High-capacity Optical Networks and Enabling/Emerging Technologies (HONET), 2015 Publication	<1 %

---

## PLAGIARISM CHECK RESULT

---

31 Muhammad Zaman Fakhar, Emre Yalcin, Alper Bilge. "IESR: Instant Energy Scheduling Recommendations for Cost Saving in Smart Homes" IEEE Access, 2022  
Tseseu Kwan Lee  
Publication <1 %

---

---

32 icdayroi.com  
Internet Source <1 %

---

---

Exclude quotes  On

Exclude matches  Off

Exclude bibliography  On

PLAGIARISM CHECK RESULT

<b>Form Title: Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)</b>			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1 of 1



**FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY**


<b>Full Name(s) of Candidate(s)</b>	Lee Kah Poh
<b>ID Number(s)</b>	19ACB06226
<b>Programme / Course</b>	CS
<b>Title of Final Year Project</b>	USING SWARM INTELLIGENCE TECHNIQUE TO OPTIMIZE THE ENERGY CONSUMPTION OF SMART HOME APPLICATION ARCHITECTURE

<b>Similarity</b>	<b>Supervisor's Comments (Compulsory if parameters of originality exceed the limits approved by UTAR)</b>
<b>Overall similarity index: <u>8</u> %</b>  <b>Similarity by source</b> Internet Sources: <u>5</u> % Publications: <u>4</u> % Student Papers: <u>0</u> %	ok
<b>Number of individual sources listed of more than 3% similarity: <u>0</u></b>	ok
<b>Parameters of originality required, and limits approved by UTAR are as Follows:</b> (i) Overall similarity index is 20% and below, and (ii) Matching of individual sources listed must be less than 3% each, and (iii) Matching texts in continuous block must not exceed 8 words <i>Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.</i>	

Note: Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

***Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.***

PLAGIARISM CHECK RESULT

  
\_\_\_\_\_  
Signature of Supervisor  
Name: Tseu Kwan Lee  
Date: 2022/09/08

\_\_\_\_\_  
Signature of Co-Supervisor  
Name: \_\_\_\_\_  
Date: \_\_\_\_\_

## FYP 2 CHECKLIST



## UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION  
TECHNOLOGY (KAMPAR CAMPUS)

## CHECKLIST FOR FYP2 THESIS SUBMISSION

Student ID	19ACB06226
Student Name	Lee Kah Poh
Supervisor Name	Ms. Tseu Kwan Lee

TICK (✓)	DOCUMENT ITEMS
	Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
	Front Plastic Cover (for hardcopy)
✓	Title Page
✓	Signed Report Status Declaration Form
✓	Signed FYP Thesis Submission Form
✓	Signed form of the Declaration of Originality
✓	Acknowledgement
✓	Abstract
✓	Table of Contents
✓	List of Figures (if applicable)
✓	List of Tables (if applicable)
	List of Symbols (if applicable)
✓	List of Abbreviations (if applicable)
✓	Chapters / Content
✓	Bibliography (or References)
✓	All references in bibliography are cited in the thesis, especially in the chapter of literature review
✓	Appendices (if applicable)
✓	Weekly Log
✓	Poster
✓	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
✓	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

\*Include this form (checklist) in the thesis (Bind together as the last page)

## FYP 2 CHECKLIST

I, the author, have checked and confirmed all the items listed in the table are included in my report.

A handwritten signature in black ink, appearing to be 'A.A.', is written on a light yellow rectangular background.

---

(Signature of Student)

Date: 26/8/2022