SMART LIGHT CONTROL WITH NOTIFICATION ALERT FOR SMALL SCALE FARM

NURHAYATI BINTI NORDIN

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

SMART LIGHT CONTROL WITH NOTIFICATION ALERT FOR SMALL SCALE FARM

NURHAYATI BINTI NORDIN

This report is submitted in partial fulfillment of the requirements for the Bachelor of [Computer Science (Computer Networking)] with Honours.

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2021

DECLARATION

I hereby declare that this project report entitled

SMART LIGHT CONTROL WITH NOTIFICATION ALERT FOR SMALL SCALE FARM

is written by me and is my own effort and that no part has been plagiarized without citations.

yadd

STUDENT: ______________________________ Date: 3/9/2021

(NURHAYATI BINTI NORDIN)

I hereby declare that I have read this project report and found this project report is sufficient in term of the scope and quality for the award of Bachelor of [Computer Science (Computer Networking)] with Honours.

SUPERVISOR: ______________________________ Date: 4/9/2021

(TS. DR. NAZRULAZHAR BIN BAHAMAN)

DEDICATION

From the bottom of my heart, I dedicated this thesis to my beloved mother, Hamidah Binti Hasan, my siblings (Norfarahin Nordin and Norhamiera Nordin), family members that always be by my side. I am so thankful for the guidance that they showed to me, the comments and suggestions over the past few years. I hope this achievement will make them happy. Not to forget to my lecturer and supervisor, Ts. Dr. Nazrulazhar bin Bahaman who always give a lot of guidance and advices throughout this journey in Universiti Teknikal Malaysia Melaka (UTeM). Last but not least to all my dear friends who have encouraged, guided and inspired me with a lot of motivations throughout this pandemic.

In memory of
My beloved late father, Nordin Bin Abu Samah.

ACKNOWLEDGEMENTS

In the Name of Allah, the Most Merciful, the Most Compassionate. All Praise to Allah SWT the Almighty for giving me the strength to finish my PSM. I am grateful to the Allah Almighty for the good health and wellbeing that were necessary can managed to complete the report of Final Year Project I & II within the time given by my supervisor, Ts. Dr. Nazrulazhar bin Bahaman. Sincerely thank you to my supervisor for the guidance and encouragement in finishing this project. I got learn so much from your advice and comment and helped me a lot into doing my best. Thank you for everything and may Allah bless you. Last but not least, I would like to express my sincere gratitude to my family and friends for the support and willingness to spend their precious time to help me complete this project.

ABSTRACT

Smart Light Control with Notification Alert form small-scale farm is a project that being developed using NodeMCU ESP8266 with the integration of IoT technology for the farmer that has small-scale farm that need a proper simple and low-cost management system to ease their farming daily activity. In this project, the user will can control and monitor their farm condition easily through the mobile application developed. The problem that they usually faced when managing the farm manually is in terms of the farm crop yields that often to damage due to the intrusion from the wild animal and also being stolen by any irresponsible anonymous person who illegally entered the farm. Hence, this can cause loss towards the farmer in the aspects of money, resources and energy in maintaining the farm condition. Next, the farmer somehow facing difficulties in control the light source in the farm area where they at one point might be hard for them to leave the work area with a troubling feeling or maybe to control their business and track it when they are out of the work region. So at least, by turning the light at the farm could reduce the uneasy feeling where people will not be easily intended to enter the farm to do damage. So, this project will use NodeMCU ESP8266 as the main component hardware to be connect with PIR motion sensor as to detect motion where it also can produce buzzer alert sound after being triggered. In addition, a literature review is required to gather all relevant information regarding the hardware, software, language, and approach employed in this project. All of the components must be appropriate for the project at hand. Then, to construct a prototype, combine all of the hardware and software. Finally, test and verify the prototype to ensure that all of the components are working together.

ABSTRAK

Kawalan Cahaya Pintar dengan Notifikasi Pemberitahuan untuk kebun berskala kecil adalah projek yang dibangunkan menggunakan NodeMCU ESP8266 dengan integrasi teknologi IoT untuk pekebun yang mempunyai kebun berskala kecil yang memerlukan sistem pengurusan kos mudah dan rendah untuk meringankan aktiviti harian pertanian mereka. Dalam projek ini, pengguna akan dapat mengawal dan memantau keadaan kebun mereka dengan mudah melalui aplikasi mudah alih yang dibangunkan. Masalah yang biasanya mereka hadapi ketika menguruskan kebun secara manual adalah dari segi hasil tanaman kebun yang sering dirosakkan akibat pencerobohan dari haiwan liar dan juga dicuri oleh mana-mana individu yang tidak bertanggungjawab yang memasuki kawasan kebun secara haram. Oleh itu, ini boleh menyebabkan kerugian kepada petani dalam aspek wang, sumber dan tenaga dalam mengekalkan keadaan kebun. Seterusnya, pekebun juga sering menghadapi kesukaran untuk mengawal sumber cahaya di kawasan kebun di mana mereka pada satu ketika mungkin sukar bagi mereka untuk meninggalkan kawasan kerja dengan perasaan yang menyusahkan atau mungkin untuk mengawal perniagaan mereka dan mengesannya apabila mereka berada di luar rantau kerja. Jadi sekurang-kurangnya, dengan menghidupkan cahaya di kebun dapat mengurangkan perasaan yang tidak selesa di mana orang tidak akan mudah bertujuan untuk memasuki ladang untuk melakukan kerosakan. Jadi, projek ini akan menggunakan NodeMCU ESP8266 sebagai perkakasan komponen utama untuk berhubung dengan sensor gerakan PIR untuk mengesan pergerakan di mana ia juga boleh menghasilkan bunyi amaran *buzzer* selepas dicetuskan. Di samping itu, kajian literatur diperlukan untuk mengumpul kesemua maklumat yang relevan mengenai perkakasan, perisian, bahasa, dan pendekatan yang digunakan dalam projek ini. Semua komponen mestilah sesuai untuk projek yang bakal dibangunkan bagi memenuhi tujuan untuk membina prototaip, menggabungkan semua perkakasan dan perisian. Akhirnya, uji dan sahkan prototaip untuk memastikan semua komponen bekerjasama.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

PAGE

LIST OF FIGURES

PAGE

LIST OF ABBREVIATIONS

FYP	**-**	**Final Year Project**
CCTV	**-**	**Closed-circuit Television**
IoT	**-**	**Internet of Things**
API	**-**	**Application Programming Interface**
GUI	**-**	**Graphical User Interface**
SMS	**-**	**Short Message Service**
RFID	**-**	**Radio Frequency Identification**
GSM	**-**	**Global System for Mobile Communication**
Wi-Fi	**-**	**Wireless Fidelity**
USB	**-**	**Universal Serial Bus**

CHAPTER 1: INTRODUCTION

1.1 Introduction

Farming is one of the usual activities that people used to fill in their leisure time as it brings a lot of benefits in terms of encouraging crop yields, productive energy, good body physical condition as it is maintaining an individual's health and many more benefits. Hence, people in the rural areas are also enjoying themselves in carrying out a small-scale farm activity. Despite a small-scale farm area, good control management is needed to ensure no harm and destructions towards their crop yields or any intrusions tragedy happening in their particular farm area.

Thus, as a result of the advancement of technology and people's dependence on smartphones to meet ever-increasing demands, it has become an easy and fast way to solve everyday life tasks. As a consequence, it is important to have a technology that can monitor applications through IoT. One of the IoT projects for low-cost solutions that can be used is Smart Light Control with Notification Alert for Small Scale Farm. Control smart lights and receive notifications Alert for small scale farm is one of the IoT projects for low-cost solutions that can be implemented specifically for small scale farmers in rural areas as an alternative to using CCTV, which is very expensive, in order to increase and improve the safety level of their farm.

In addition, this is an alternative approach to control, which is that accepting data from the sensors could give the farmers at least a small picture of what was happening on their farm. Hence, storing all of the data could help them analyze if there are any needed improvements.

1.2 Problem Statement

Smart Light Control with Notification Alert for Small Scale Farm is a project that enables users to ease their management activities in order to improve and maintain secure security levels for small-scale farms in particular, especially in rural areas, using a low-cost medium solution. The Internet has become a part and portion of human life with the exponential increase in the number of internet users over the past decade. This system can control the light and also detecting motion through the sensors for the specified farm area. Therefore, the established project consists of two parts, including a simple API, hardware and software such as devices, NodeMCU ESP8266 board, PIR sensor and many more.

This project approach in order to help ease some problems faced by the users, such as the high amount of Internet bandwidth consumption when using CCTV which is required to load the acquired footage videos from the CCTV where particularly might be very costly, especially for a small single scale farmer. In addition, there are more workloads needed to be hired, like the extra workers to facilitate the farmer in tracking for any intrusion issues at the specified working area. Thus, it might be hard for them to leave the work area with a troubling feeling or maybe control their business and track it when they are out of the work region.

Table 1.1 Summary of Problem Statement

PS	Problem Statement
PS1	CCTV installation for a single-scale farmer is quite costly and high-maintenance in order to manage the farm which is not worth it.

1.3 Project Question

Once the architecture is understood, how can this project solve the scenario of manually manage the small-scale farm? The project questions are as follows:

Table 1.2 Summary of Problem Questions

PS	PQ	Problem Questions
PS1	PQ1	What type of system will be developed?
	PQ2	How efficient is the prototype's sensor?
	PQ3	How to give a notification alert to the farmer?

1.4 Project Objectives

Based on project questions, this project consists three main objectives to ensure that the prototype can be fully implemented. Table 1.3 below is a summary of the project objectives.

Table 1.3 Summary of Project Objectives

PS	PQ	PO	Project Objective
PS1	PQ1	PO1	To develop a low-cost IoT solution prototype that can automatically detect motion based on light control
	PQ2	PO2	To notify the user through notification alert as a precaution
	PQ3	PO3	To test the functionality of the prototype

1.5 Project Scope

For this project, there are several scopes that are going to be focused on:

1) User: any small-scale farmer that needs a system to guard their farm

2) System: Develop a simple system for the user to manage their farm through light control with hardware utilization like NodeMCU ESP8266, led, PIR motion sensor and others.

3) The functionality of automatic on/off light primarily based totally on the movement sensor that stumbles on the movement presence of wild animals or intruders and also light brightness control

4) Real-time notification alert to let the user know the condition of their business area when detecting any motion indicates the light is automatically on in the specified area.

5) Implementing some modules like report modules for graph analysis

1.6 Problem Contribution

This prototype is designed especially for people with a small-scale farm that need a low-cost solution approach in handling and keeping track of their farm state, which is a better solution in saving money, time, and resources. Table 1.4 below shows the summary of the project contribution.

Table 1.4 Project Contribution

PS	PQ	PO	PC	Project Contribution
PS1	PQ1	PO1	PC1	It proposed to develop a simple API system where users can easily monitor and control their farm conditions.
	PQ2	PO2	PC2	It proposed a prototype that able to strengthen the security level of a single small-scale farm with hardware and software utilization.
	PQ3	PO3	PC3	It proposed an appropriate notifications alert that notifies the user about the farm condition.

1.7 Report Organization

This report is divided into seven chapters: Chapter 1 contains the Introduction, Chapter 2 is for Literature review, Chapter 3 consists of Methodology, Chapter 4 is for the Design, Chapter 5 contains the Implementation, Chapter 6 contains the Testing and lastly, Chapter 7 for Conclusion.

Chapter 1: Introduction

This chapter will focus on the introduction, problem statements, project questions, project objectives, project scope, project contribution, report organization and conclusion.

Chapter 2: Literature review

This chapter will focus more on related or previous work on this project and a critical analysis of the current problem, reasoning, and suggested solution. This project's explanation and details are assisted by reading materials and a conference paper.

Chapter 3: Project Methodology

This chapter describes the methodology that will be used in this project as well as the project milestones.

Chapter 4: Analysis and Design

This chapter will deliberate over the project's problem analysis, requirement analysis, high-level design, database design, and comprehensive design.

Chapter 5: Implementation

This chapter will discuss the setup of a software development system, software configuration management, and the status of implementation.

Chapter 6: Testing

Beginning with microcontrollers and applications, Chapter 6 will begin the application testing and debugging phase. At this point, the findings are compared to the objectives in order to draw conclusions and make assumptions.

Chapter 7: Project Conclusion

For chapter 7 it will discuss the project summarization, project contribution, project constraints, and future work.

1.8 Conclusion

In conclusion, this chapter consists of the details and information of the proposed project, the smart light control with notification alert for small-scale farmers. Hence, all the primary objectives and also the scopes have been discussed. So, the next chapter will be proceeded and focus on the literature review of the project.

CHAPTER 2: LITERATURE REVIEW AND PROJECT METHODOLOGY

2.1 Introduction

This chapter will discuss the project's literature review in which there will be a few vital explanations that will be applied in this project. In addition, some other aspects will be discussed in this chapter involving the current issue, justification, and the solution approach for enhancement of the project.

Hence, this chapter will briefly explain and focus more on the functionality of the project prototype and the creative workflow of the project process. Thus, the detailed information is indeed vital in order to carry out this project. On the other hand, this chapter involves a few particularized discussions of some journal articles or any research done by others. It is more or less related to this current project, such as the terms of its components type and the methodology used in carrying out the project implementation.

In addition, this literature review aims to summarize all the information details that can be used to enhance the process of generating ideas. On the other side, the completed project can be used as a model for improving the current project in accordance with its intent and objectives.

2.2 Related Work

On Internet of Thing or to be known as "IoT" refers to the objects network or devices containing an embedded technology that help enable people to communicate and interact with one another. Thus, this connectivity is interrelated between the physical devices and could happen anywhere, everywhere and anytime such as whether in our home, office and many more places. In addition, it somehow can include appliances such as light fixtures or any of the devices that we owned that connected directly to the web or any other devices in some way. These devices also include smart gadgets where can be accessed and controlled from a remote location which benefits the users from a wide range of industries.

The common situation that the small-scale farmer has faced is they usually use to have the unease feelings and worry most of the time thinking about their farm condition as they left the place. One of the reasons is the presence of wild animals or any intruders that illegally enter the farm area, which either might cause damage or steal all the farm crop yields that can make a huge loss in terms of money, resources, or any other factors. The figure below shows the taxonomy flow of the smart light control with notification alert for small-scale farm

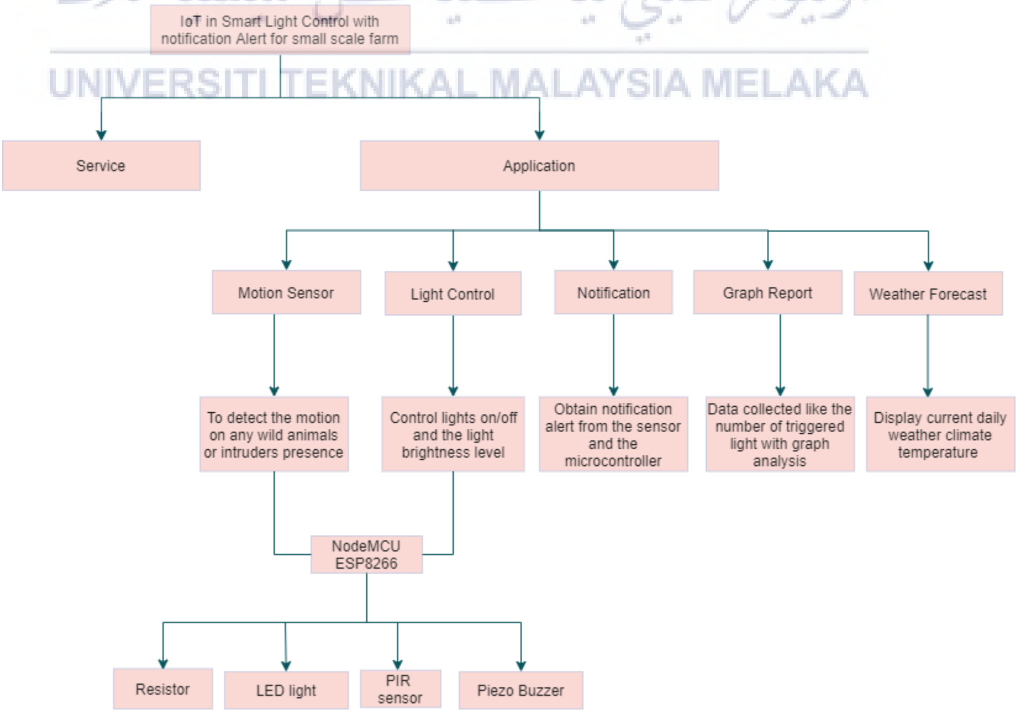

IoT in Smart Light Control with notification Alert for small scale farm
Service
Application
Motion Sensor
Light Control
Notification
Graph Report
Weather Forecast
To detect the motion on any wild animals or intruders presence
Control lights on/off and the light brightness level
Obtain notification alert from the sensor and the microcontroller
Data collected like the number of triggered light with graph analysis
Display current daily weather climate temperature
NodeMCU ESP8266
Resistor
LED light
PIR sensor
Piezo Buzzer

Figure 2.1: Taxonomy Structure

The taxonomy figure shown above describes the relation of the IoT in smart light control with notification alert with some derivation that involves several main components and explanation of its functionality. This taxonomy explains the prototype division aspect as an overview or guideline in developing this project and ensuring that this prototype development is on the right track. Therefore, for this development process it will focus more on the application to be developed for the targeted user that consists three main key components that are determined as the foundation for this project's development which it also has four functionality such as the motion detection using PIR sensor, light control, notification and also graph report.

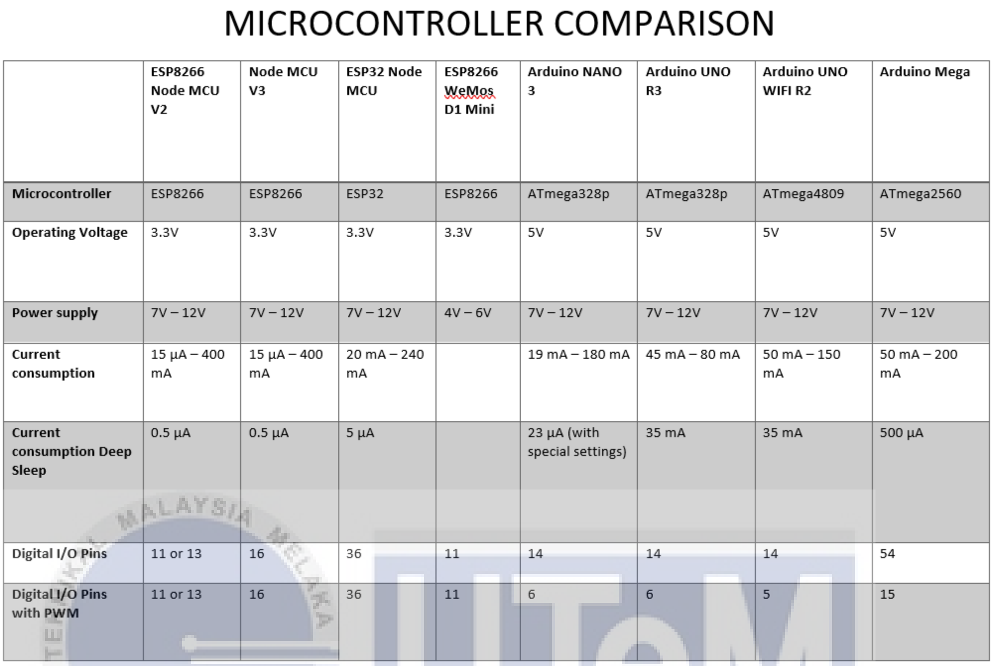
2.2.1 NodeMCU ESP8266

NodeMCU ESP8266 is module hardware that the community has widely used to act as a microcontroller, especially in the IoT world. Hence, it’s also designed to develop and build digital devices and interactive objects that can sense and control objects through both physically and digitally. ESP8266 is also one of the affordable and reasonable microcontrollers nowadays that fits users well, especially students undergoing any IoT environment research project.

The MCU within the ESP8266 is primarily used to control and process data, which can be acquired and processed via the various communications ports and data uploaded via the Wi-Fi module linked to the Internet. The most significant benefit of employing this NodeMCU ESP8266 is its low cost. (T. Qiang et al., 2018).

Besides, several types of other microcontroller boards can be used in order to build a certain type of project such as the ESP32, Arduino Uno, Arduino, Arduino WIFI R2, Arduino Leonardo, Arduino Mega, Arduino Due, Arduino Red Board, Arduino Shields, Arduino Ethernet REV3, Arduino MKR Zero any many more. Despite all of this board being categorized under Arduino, every board consists of different functionality and features, such as the total number of pin slots, whether digital or analog, the USB port, and other aspects.

Table 2.1 Microcontroller Comparison



MICROCONTROLLER COMPARISON

	ESP8266 Node MCU V2	Node MCU V3	ESP32 Node MCU	ESP8266 WeMos D1 Mini	Arduino NANO 3	Arduino UNO R3	Arduino UNO WIFI R2	Arduino Mega
Microcontroller	ESP8266	ESP8266	ESP32	ESP8266	ATmega328p	ATmega328p	ATmega4809	ATmega2560
Operating Voltage	3.3V	3.3V	3.3V	3.3V	5V	5V	5V	5V
Power supply	7V – 12V	7V – 12V	7V – 12V	4V – 6V	7V – 12V	7V – 12V	7V – 12V	7V – 12V
Current consumption	15 µA – 400 mA	15 µA – 400 mA	20 mA – 240 mA		19 mA – 180 mA	45 mA – 80 mA	50 mA – 150 mA	50 mA – 200 mA
Current consumption Deep Sleep	0.5 µA	0.5 µA	5 µA		23 µA (with special settings)	35 mA	35 mA	500 µA
Digital I/O Pins	11 or 13	16	36	11	14	14	14	54
Digital I/O Pins with PWM	11 or 13	16	36	11	6	6	5	15

Hence, NodeMcu is also a microcontroller that is based on open-source programming and rigging progression framework based on the ESP8266, a small System-on-a-Chip that is cheap and can implement much functionality. In addition, NodeMCU ESP8266 comes up with 16 GPIO pins that are also integrated with its software platform, which is known as Integrated Development Environment (IDE), the main Arduino development program. It uses some programming language codes such as C and C++ for the IDE where it also has 4 MB of flash memory, a clock speed of 80-160 MHz, roughly 50 KB of useable RAM, and an on-chip Wi-Fi transceiver.

Figure 2.2: NodeMCU ESP8266 V2 board Hardware

2.2.2 Sensor Detection

One feature that makes an Arduino so beneficial is the ease with which it can obtain sensor values. There are so many types of sensors that can be used with Arduino according to some particular aspects such as the ultrasonic sensor, humidity sensor, speed sensor, gas sensor, photoresistor sensor and so on so forth. In this project, Passive Infrared (PIR) motions sensor is being used based on its reliability in detecting motions and large objects such as human, animals, and other moving objects with 99% accuracy. This detection process is happening through infrared light to define or determine the presence of the object.

Figure 2.3: PIR motion sensor

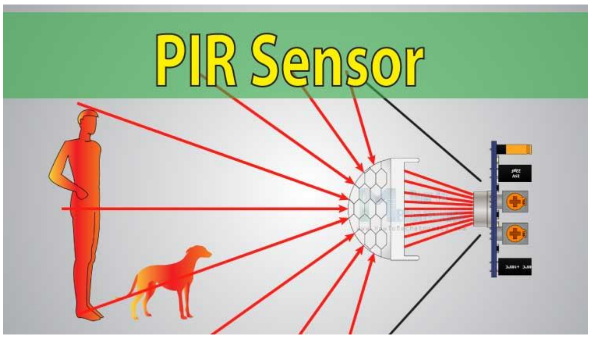


Figure 2.4: Sensing activity details overview

2.3 Critical Review of the current problem and justifications

2.3.1 Introduction

IoT is the most general topic or essence in every aspect due to its factors that ease most people in this world's work or routine. Hence, it is also reflecting the agriculture environment and day goes by the expansion of IoT technology had changed the perspective of handling and managing this sector. Even though few inventions can ease this job, some issues still arise, such as the intrusion and trespassing problem that shown that they still lack security level consideration in inventing a product.

According to (A. Parmar et al., 2020) research in the journal, the interference of animals in agricultural lands causes a huge crop loss. Crop damage due to raiding wild animals has become a major issue of concern these days. Animals like wild boars, macaques, porcupines, deer, monkeys and bears are extremely destructive and have also caused human casualties sometimes. The total losses in crop yield are high for potatoes and wheat in villages. The small farmers have to give up in a range of 40 until 50 percent of their crop yields to the untamed animals and they can’t even take any action or rights regarding to the firm wildlife laws.

2.3.2 Importance of project

The aim of this project is actually to improve the small scale farm security standard which is somehow people might think that it is not important to keep eyeing or safeguard for a small farm area with proper management in facts these are the small things that people need to emphasize and give enough attention to it in order to improve the agriculture sector which giving benefits to a small farmer that might help in generating the production of quality plants and at the same time can provide a profit on the yield grown. Hence, the importance of this project is to ensure that users would be able to receive notification alerts when there are any motions detected whether it is humans that try to trespassing into the farm or it also might be any entry of wild animals that can damage the crop yields. With the help of the microcontroller, sensor and other components it may help to lessen the difficulties in handling their farm.

2.3.3 Previous Existing Product

The previous existing product consists of its methodology that explains the system's working flow, particularly according to the flowchart, giving a clear vision about the whole concept for the project. Each of the methodologies will differentiate the product one to another where it can be used as analysis in developing this project.

2.3.4 SMS Notifications On-Off Room Lights with Body Detection Using Microcontrollers

This previous project aims to sense the movement or motion made by the human body specifically in the room area through light control with the implementation of the light sensor and the PIR sensor. In addition, with the intensity detection of light in the room is reduced if there is any human body motion that indicates people are entering the room.

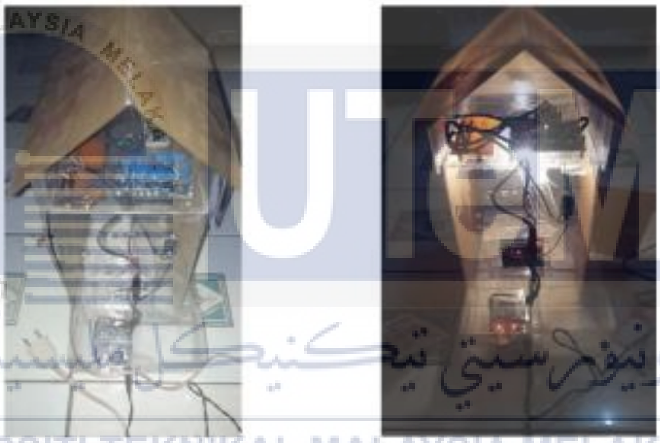


Figure 2.5: Example prototype of the project

Figure 2.5 shows the example prototype made to show the room environment when it detects movement from the human body that will activate the sensor and instantly turn the lights on.

2.3.5 Smart Intrusion Detection System for Crop Protection by Arduino

For this project that is wise Intrusion System for Crop protection is one amongst the helpful project inventions that is integrated with a ton of elements that build it a complete product. With the mixing of few devices like the ultrasonics sensor, the PIR device and conjointly few buzzers connected that represent an enormous main half within the system. This analysis aims to emphasize a secure system to safeguard the farmland, notably hindrance from animals like elephants, which may be able to destroy the crops within the farmland space.

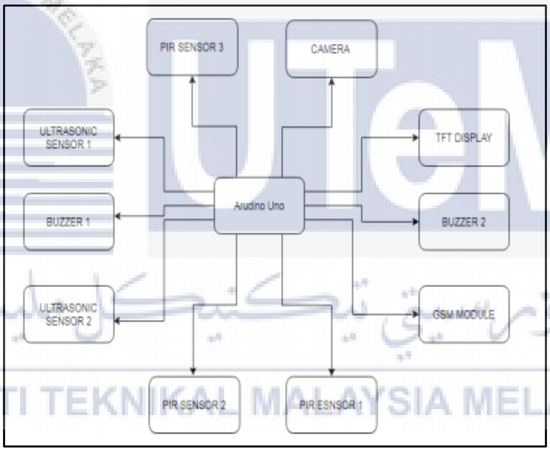


Figure 2.6: Block Diagram

Figure 2.6 shows the block diagram used in this research project where it is shown that there are many components used and connected to the heart of the system, the Arduino Uno circuit board.

2.3.6 Smart farmland using Raspberry Pi Crop Prevention and Animal Intrusion Detection System

Next, for this project research according to (Bindu D and Dilip Kumar M D et al. 2018) saying that over the years, animals from the protected area have repeatedly attacked the crop field, and the crop field's security has become a significant concern. Thus, in this journal, they present a realistic technique to ward them off in this article by developing a device that studies the animal's behavior, detects the animal, and generates various sounds that irritate the animal while also alerting the author.

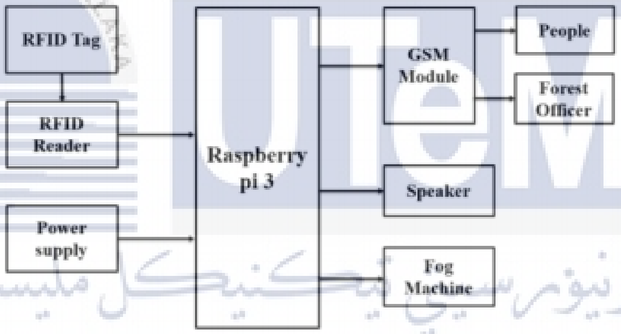


Figure 2.7: Functional Block Diagram

Figure 2.7 shows the functional block diagram used in the research above where it shows the flow concept of the prototype that consists of components such as the RFID tag, RFID reader, GSM Module, speaker and also the main item components which are the Raspberry Pi board as the heart of the system. Then, it will transmit the data received by the RFID tag to the targeted users through a message (SMS) notification on their phones.

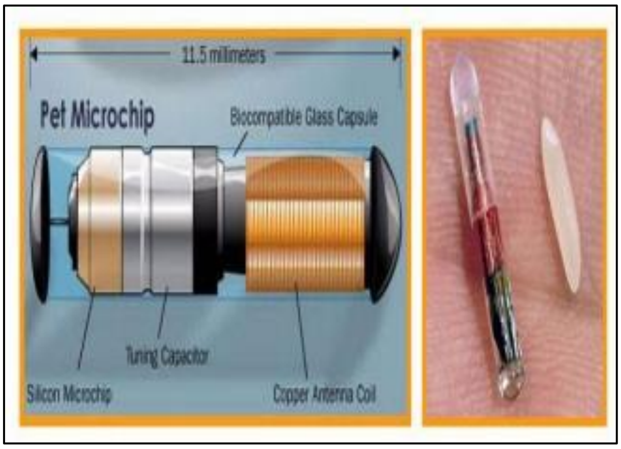


Figure 2.8: RFID Pet Microchip

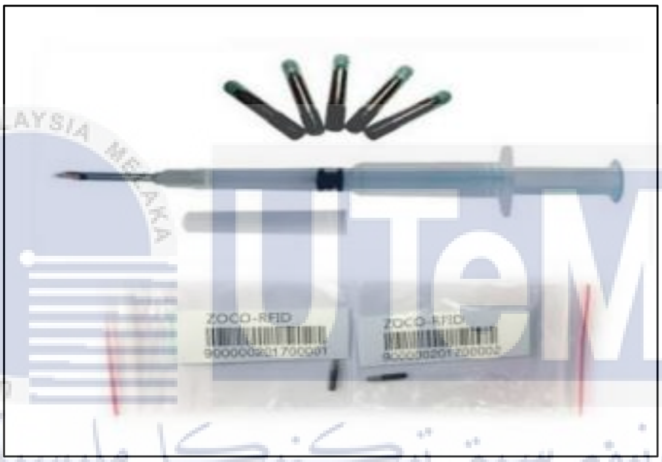
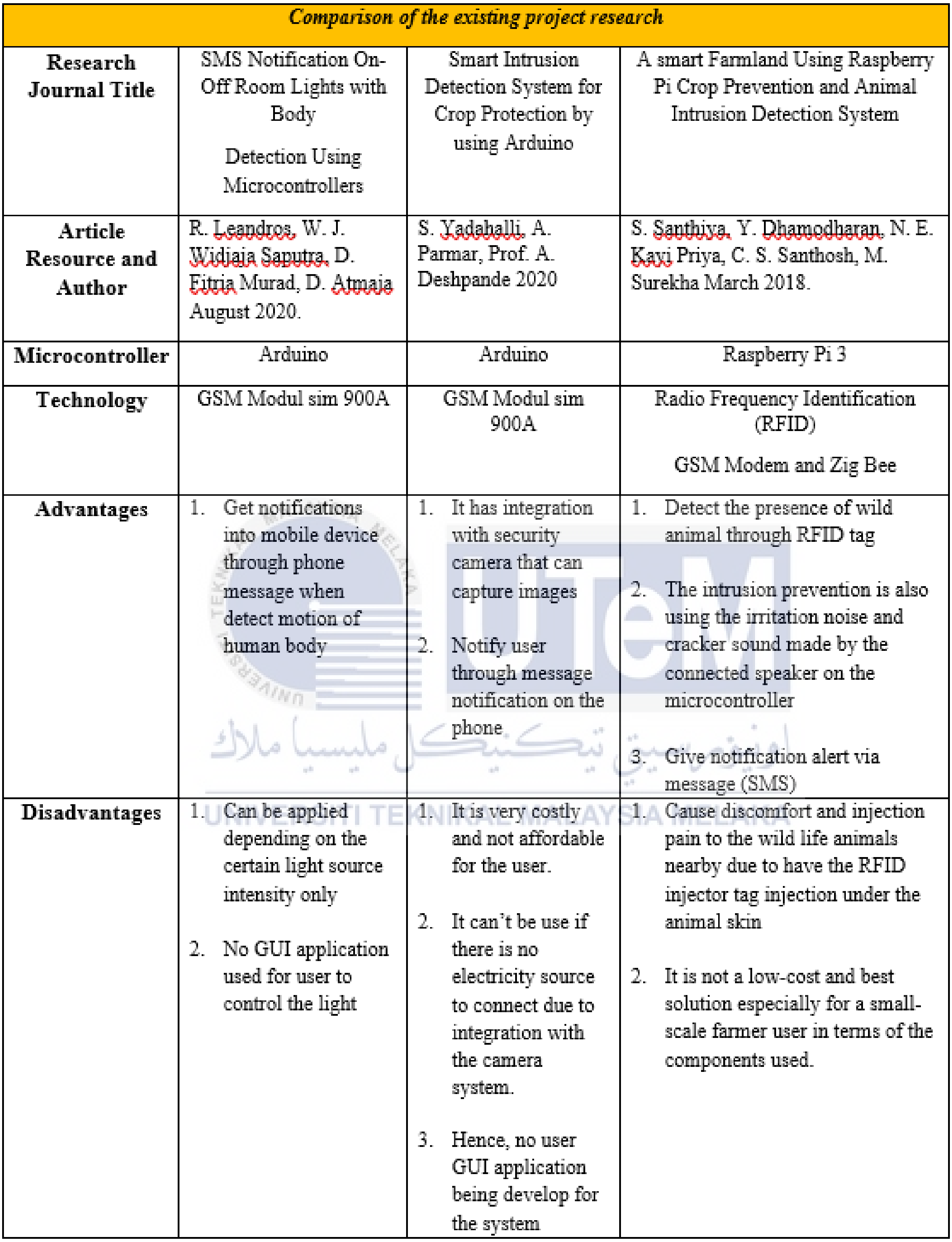


Figure 2.9: RFID Injector

Figure 2.8 above shown the RFID Pet Microchip and figure 2.9 displays the RFID injector used. Both of these two items are applied to the animal where the RFID tag will be injected under their skin by the RFID Injector, which might facilitate them in tracking the wild animal's presence, especially into the farmland as they already do the injection. Hence, the RFID reader will then read and detect if there might be any animals that are closer to the particular area and send the SMS by using the GSM module to the targeted user.



Comparison of the existing project research			
Research Journal Title	SMS Notification On-Off Room Lights with Body Detection Using Microcontrollers	Smart Intrusion Detection System for Crop Protection by using Arduino	A smart Farmland Using Raspberry Pi Crop Prevention and Animal Intrusion Detection System
Article Resource and Author	R. Leandros, W. J. Widjaja Saputra, D. Fitria Murad, D. Atmaja August 2020.	S. Yadahalli, A. Parmar, Prof. A. Deshpande 2020	S. Santhiya, Y. Dhamodharan, N. E. Kavi Priya, C. S. Santhosh, M. Surekha March 2018.
Microcontroller	Arduino	Arduino	Raspberry Pi 3
Technology	GSM Modul sim 900A	GSM Modul sim 900A	Radio Frequency Identification (RFID) GSM Modem and Zig Bee
Advantages	1. Get notifications into mobile device through phone message when detect motion of human body	1. It has integration with security camera that can capture images 2. Notify user through message notification on the phone	1. Detect the presence of wild animal through RFID tag 2. The intrusion prevention is also using the irritation noise and cracker sound made by the connected speaker on the microcontroller 3. Give notification alert via message (SMS)
Disadvantages	1. Can be applied depending on the certain light source intensity only 2. No GUI application used for user to control the light	1. It is very costly and not affordable for the user. 2. It can't be use if there is no electricity source to connect due to integration with the camera system. 3. Hence, no user GUI application being develop for the system	1. Cause discomfort and injection pain to the wild life animals nearby due to have the RFID injector tag injection under the animal skin 2. It is not a low-cost and best solution especially for a small-scale farmer user in terms of the components used.

Table 2.2: Comparison of existing project research

2.4 Proposed Solutions

In this chapter, the proposed solution will discuss the way or the best approach that can be made in improvising this project where all the previous concept in the discussed articles journal will be taken into account. Hence, the development of this low-cost prototype might be helpful to the users which are the small-scale farmers out there, in managing their farm much easier than previous manual management routine that they used to. In addition, it can also save time, resources and energy along with the notification alert that can notify them in real-time.

Therefore, this project approach uses NodeMCU esp8266 as the main microcontroller that acts as the heart of the system with the WiFi package technology that will be implemented together. Hence, it will also use the PIR sensor as the component used to detect motions of any human body or any wild animals trespassing by the farm. So, when any movements are detected, it will trigger the LED lights to turn on and turn off if otherwise. Next, the NodeMCU esp8266 will be used in order to retrieve data value from the sensor and notify the user through the notification alert and use the Android application as the control platform.

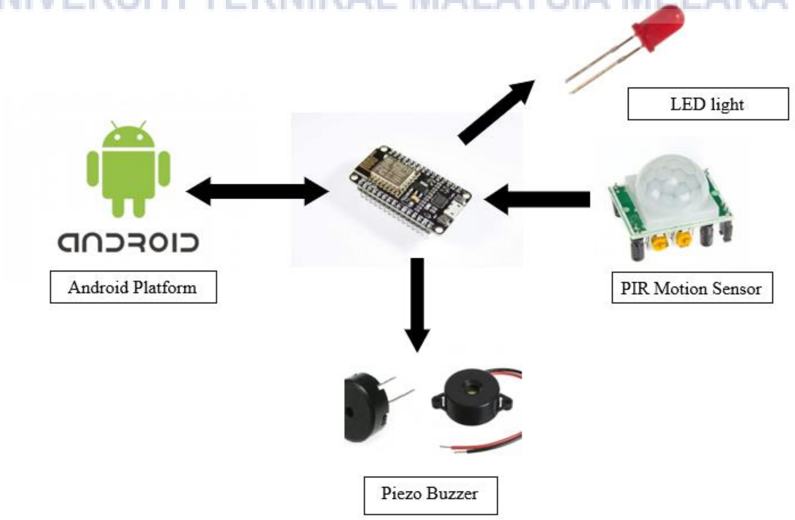


Figure 2.10: Proposed Solution

2.5 Conclusion

In conclusion, this chapter focuses more on reviewing the literature review from multiples research articles that is very useful in enhancing and being the guidance in developing this prototype. Hence, the articles discussed in this chapter will be one of the references for this project to make any suitable improvements in the future. Thus, the project methodology will be discussed in the next chapter that will explain Smart light control with notification alert for small-scale farm project planning and discussing few related requirements.

CHAPTER 3: PROJECT METHODOLOGY

3.1 Introduction

This chapter discusses concerning the project methodology and project milestones. In project methodology, the technique utilized in this project are justified and project milestones were created to follow the point in time of the project so that the project can be complete. A project methodology ought to be chosen earlier throughout the project coming up with. The approach depends on the formalized necessities and each final project set up. The approach will affect all of the looks, so it plays an enormous half in developing this method as a foundation of the system.

In addition, the project process model is an abstract representation of a project process. Each process model represents a process from a particular perspective and thus, provides only partial information about the process. Hence, this chapter helps arrange the project planning as it also outlines each step for the project where it is very important to always keep up with the project development. In addition, a project milestone and Gantt chart will be included in this chapter to assist the student in fully guiding the timeline of each move. This project milestone and a Gantt map for the remainder of the project would yield the ideal outcome.

3.2 Methodology

In order to complete the project effectively with the proper methods and flows, project methodology is very important in the development phase. This subject will identify and clarify the most appropriate methodology.

This chapter will go through the four phases of the project: the Literature Review, Project Design and Component Selection, Development and Simulation, and Testing and Verification. Hence, each stage must be completed before moving on to the next stage.

Figure 3.1 shows each step in creating an effective prototype and hardware creation, including a connection between hardware and software. These measures are crucial to always being on the track and guidance that will help develop the project.

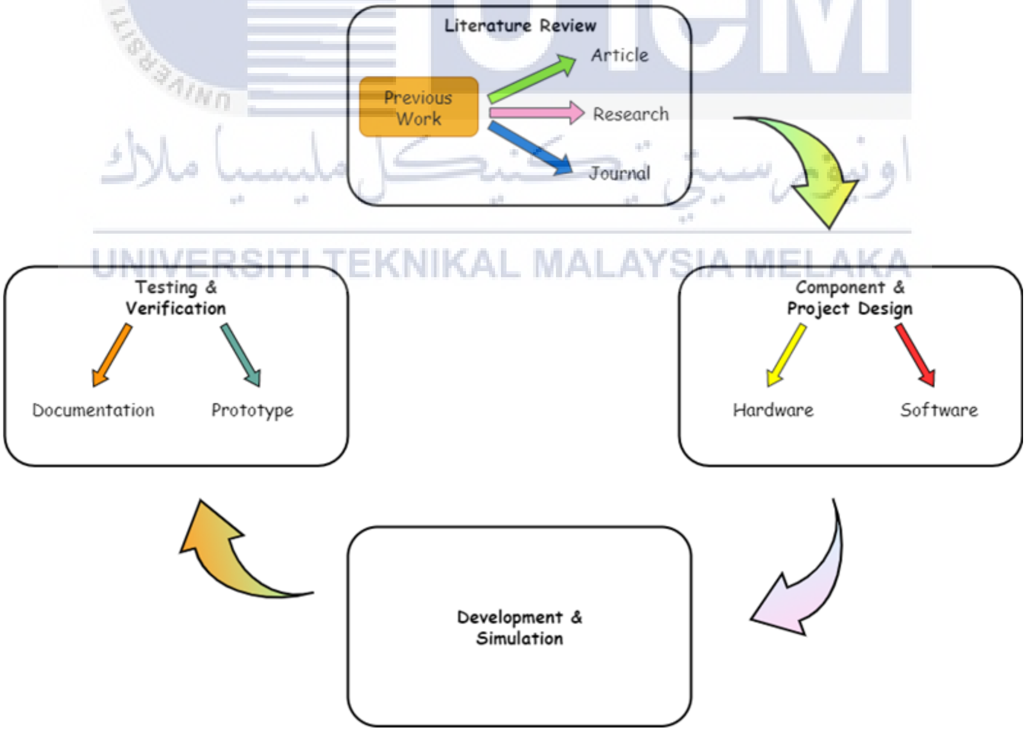


Figure 3.1: Methodology block diagram

3.2.1 Literature Review

The use of this Literature Review is to review and examine some of the previously existing work related to various sources such as the books, journals, articles, or any research papers to comprehend the concepts of the project that must be understood in completing this project. In addition, the Literature Review will facilitate in determining which hardware, software, processes and workflows that are appropriate for this project. Furthermore, it will also include a lot of information and skills.

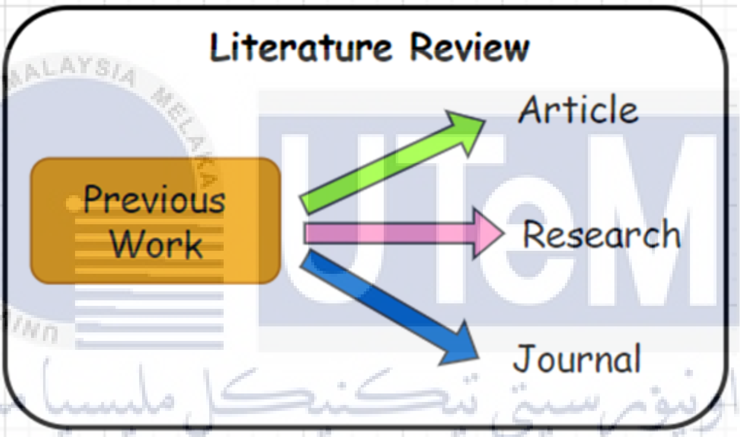


Figure 3.2: Literature Review

Based on figure 3.2 above it shows the components related that facilitate in making the reviews and examination of previously existing works in the literature review phase. So, that it will ease the design process, components identification selection and many more aspects.

3.2.2 Component and Project Design

In Component and Project Design phase, it contains few parts that will be chosen and the design will be visualized to obtain an output during this process. Hence, the components are divided into two categories which are the hardware and also software. Then, the next step will be to decide the hardware and software based on the concept proposed. This process, component and project design, aims to ensure that the design meets the goal and can solve the problem.

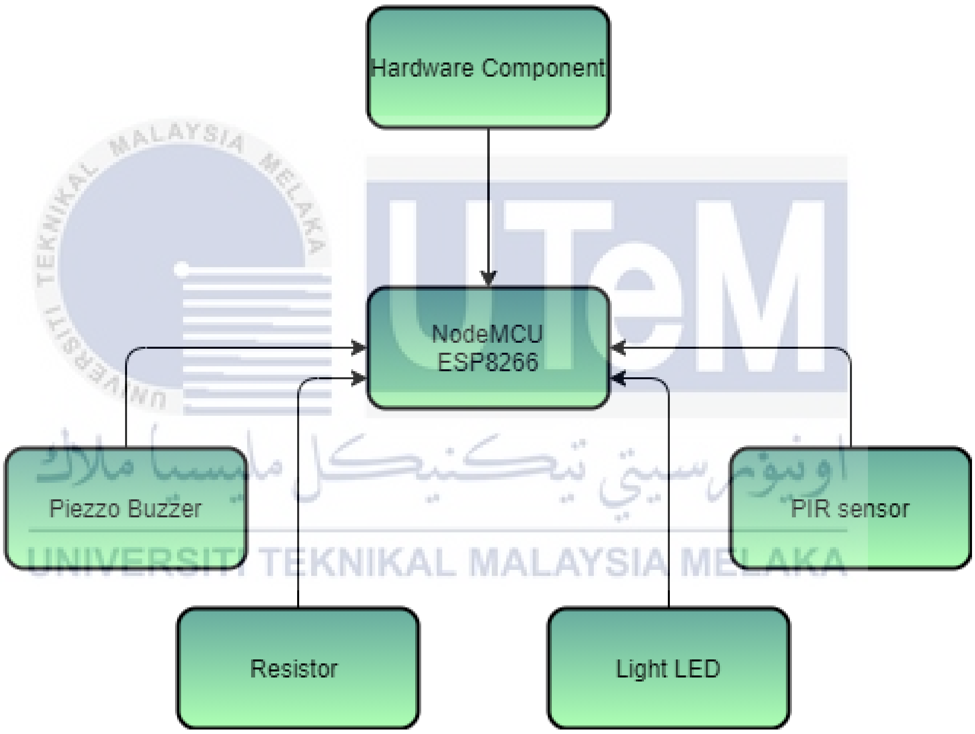


Figure 3.3: Main Hardware Component

Next, figure 3.3 refers to the summarization of the main hardware components used in this project. Each component plays its functionality respectively to ensure that the communication between the hardware and software at the end of the outcome is working smoothly.

Next, for software applications it consists few aspects that being used in completing the project and the hardware modules must be compliant with the software applications to ensure that they can be function well. The user should be able to receive notifications alerts from the app. The description of the Software Application is shown in Figure 3.4 below:

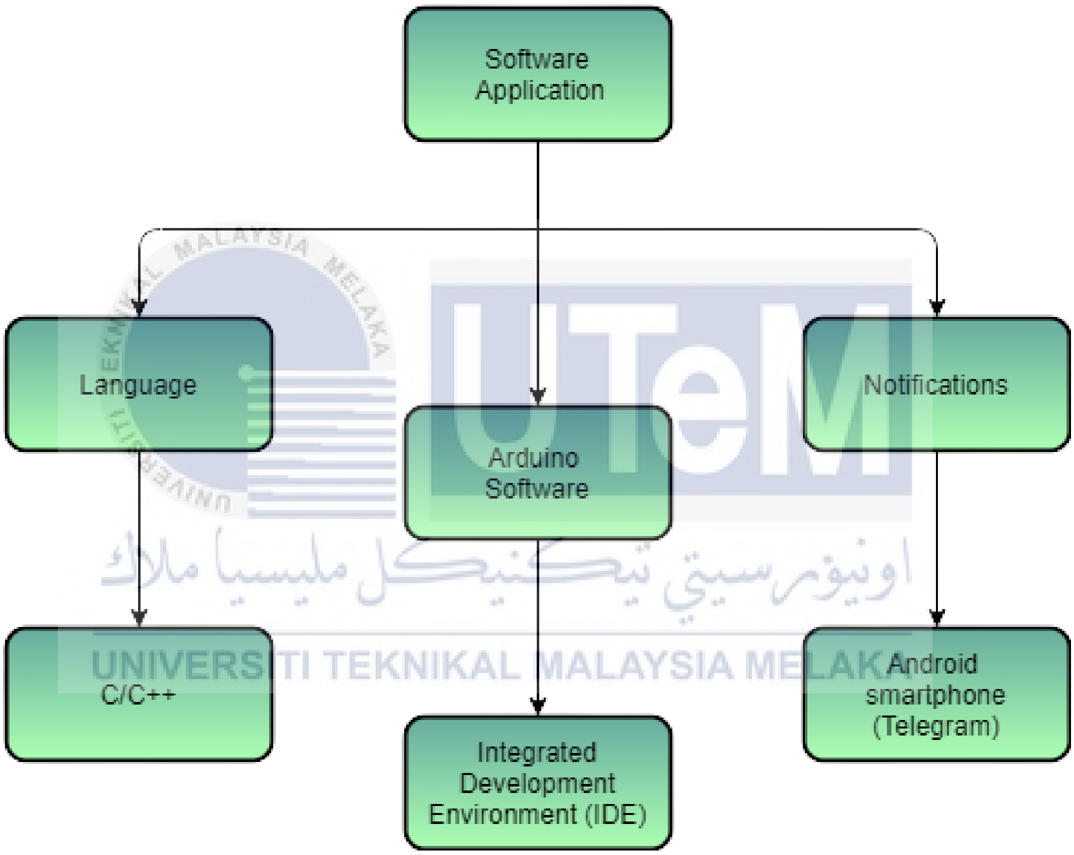

Software Application
Language
Arduino Software
Notifications
C/C++
Integrated Development Environment (IDE)
Android smartphone (Telegram)

Figure 3.4: Software Application

For figure 3.4 it is referring to the summarization of the software application of the project involving a few components aspect. Generally, it describes the language used in developing the prototype which is C/C++ language and uses Android as the medium platform of the application that can give notification alerts to the users.

3.2.3 Development and Simulation

Next, the development and simulation phase will describe and explain the development flow process with all the selected hardware and software components discussed previously. In addition, this phase will start with the hardware part development according to the schematics diagram with the software implementation.

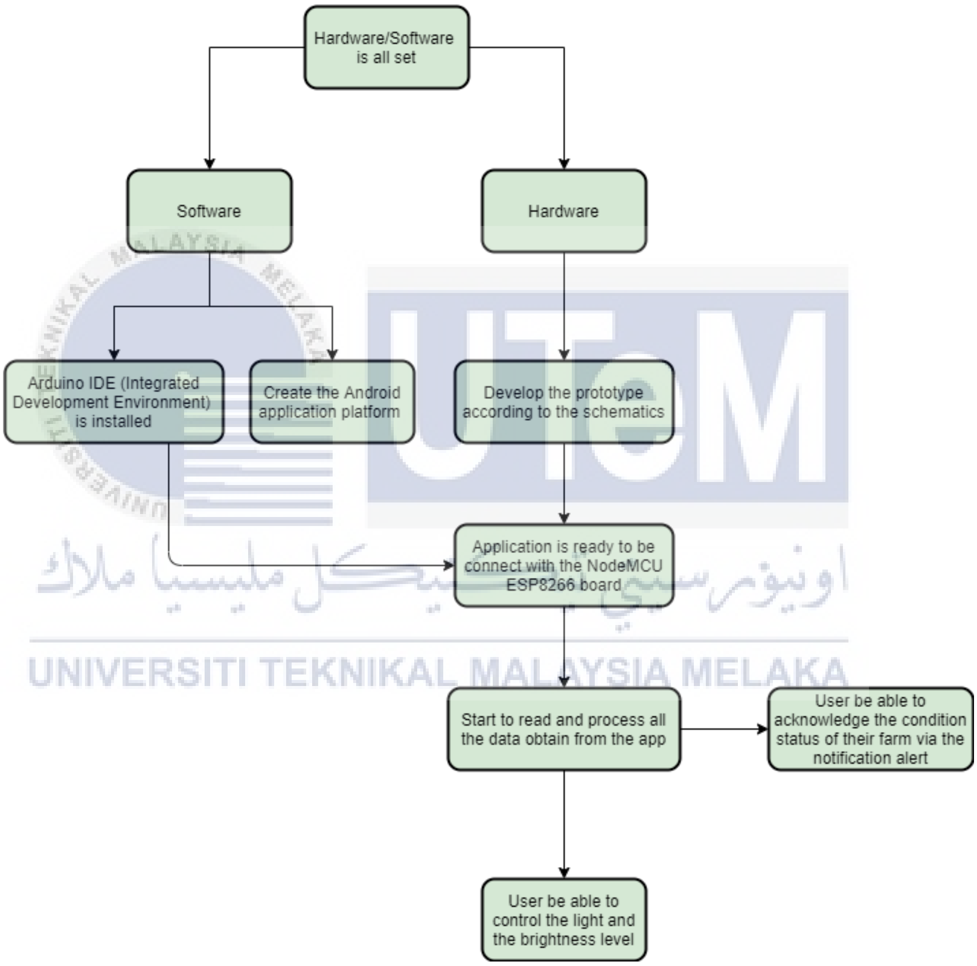

Hardware/Software is all set
Software
Hardware
Arduino IDE (Integrated Development Environment) is installed
Create the Android application platform
Develop the prototype according to the schematics
Application is ready to be connect with the NodeMCU ESP8266 board
Start to read and process all the data obtain from the app
User be able to acknowledge the condition status of their farm via the notification alert
User be able to control the light and the brightness level

Figure 3.5: Development Project Process

Figure 3.5 depicts the flow of the prototype development process, which includes both hardware and software parts. This aids in comprehending the essence of the project by serving as a guideline for the development of the project.

3.2.4 Testing and Verification

Testing is the process of running an application or program to find potential software flaws. It also aims to assist in the testing of the product to ensure that it functions as intended. This process is divided into two categories: prototype and documentation. The prototype and documentation will be the focus of testing and verification. After the prototype is complete, it will then be reviewed to ensure that the project meets the goals requirement features and functions properly. Hence, the project must be reported in a report to ensure that it is completed.

3.3 Project Milestone

A project milestone is a time period that describes the actions planned before the project's completion and the activities stage by stage. Plan Milestones serve as a point of reference for tracking project progress and identifying important project activities. Hence, in order to ensure that all project activities are completed within the project's timeframe, a project milestone will be established and scheduled. The Gantt chart will monitor the progress of each chapter to ensure that all tasks are completed within the time frame.

Table 3.1 PSM 1 Milestone

Week	Activity
W1 15 March → 21 March	Proposal PSM: Discussion
	Proposal Assessment & Verification
W2 22 March → 28 March	Proposal Correction/Improvement
	List of Supervisor/Title
W3 29 March → 4 April	Proposal Presentation & Submission via PSM ULearn
	Chapter 1 (System Development Begins)
W4 5 April → 11 April	Chapter 1

W5 12 April → 18 April	Chapter 2
W6 19 April → 25 April	Chapter 2 Project Progress
W7 26 April → 2 May	Chapter 3
W8 3 May → 9 May	Deliverable Chapter 3
W9 10 May → 16 May	*MID SEMESTER BREAK*
W10 17 May → 23 May	Chapter 4 Project Progress
W11 24 May → 30 May	Project Demonstration
W12 3 May → 9 May	Project Demonstration PSM 1 Report
W13 7 June → 13 June	Project Demonstration PSM 1 Report
W14 14 June → 20 June	Project Demonstration
W15 21 June → 27 June	Final Presentation Submission of the PSM1 Report onto the PSM ULearn
W16 28 June → 4 July	REVISION WEEK Correction on the draft report based on the comments by the Supervisor and Evaluator during the final presentation Session. Submit PSM1 Logbooks to PSM e-Repository online System. Submission of overall marks to PSM/PD committee
W17 & W18 5 July → 18 July	FINAL EXAMINATION WEEKS

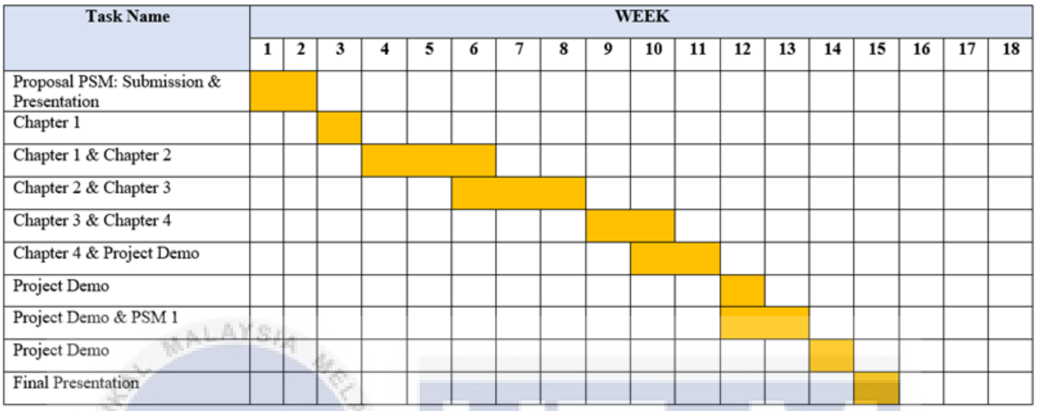
Table 3.2: PSM 2 Milestone

Week	Activity
W1 19 July → 25 July	Chapter 5
W2 26 July → 1 August	Chapter 5 Project Progress
W3 2 August → 8 August	Deliverable Chapter 5
W4 9 August → 15 August	Chapter 6 Project Progress
W5 16 August → 22 August	Deliverable Chapter 6 Chapter 7
W6 23 August → 29 August	Deliverable Chapter 7 Project Demonstration PSM2 Draft Report
W7 30 August → 5 September	Final Presentation Project Demonstration Submission of the PSM1 Report onto the PSM ULearn
W8 6 September → 12 September	FINAL EXAMINATION WEEKS Correction on the draft report based on the comments by the Supervisor and Evaluator during the final presentation Session. Submit PSM2 Logbooks to PSM e-Repository online System. Submission of overall marks to PSM/PD committee
W9 13 September → 19 Sepetember	INTER-SEMESTER BREAK Submission of the final complete report, updated & corrected PSM2 report, onto the PSM2 ULearn

3.4 Gantt Chart

Gantt Chart is used to ensure that all tasks are completed on time. It also serves as a project completion guide. Each procedure in a Gantt Chart triggers the execution of another procedure. The following table, Table 3.2, indicates the Gantt Chart.

Table 3.3 Gantt Chart



Task Name	WEEK																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Proposal PSM: Submission & Presentation	🟨	🟨																
Chapter 1			🟨															
Chapter 1 & Chapter 2				🟨	🟨	🟨												
Chapter 2 & Chapter 3						🟨	🟨	🟨										
Chapter 3 & Chapter 4									🟨	🟨								
Chapter 4 & Project Demo										🟨	🟨							
Project Demo												🟨						
Project Demo & PSM 1												🟨	🟨					
Project Demo														🟨				
Final Presentation															🟨			

3.5 Conclusion

In conclusion, the project milestones ensure that tasks are completed on schedule and in the correct manner. So, the next chapter, which is the Design chapter, will discuss the model creation and next step process flow for the system that must be completed well to meet the goal.

CHAPTER 4: ANALYSIS AND DESIGN

4.1 Introduction

This chapter explains the hardware and software specifications for IoT-based particularly for this project. Since certain mistakes can disrupt the entire project process, the design phase is critical. When creating and describing the concept and process architecture, it is critical to be careful and provide the best solutions and ideas. The project's execution will demonstrate how and why the project will be successful.

4.2 Problem Analysis

The problem analysis of this project is as stated previously in chapter 1. The farmer tends to face some difficulties such as intrusion issues from the outside or maybe wild animals to manage their small-scale farm, especially when outside the work region. Pertaining to this problem, people usually tend to install CCTV to facilitate tracking their farm condition, but CCTV installation might cost a quite expensive amount of budget to set up. It has high maintenance to keep up, including a high amount of bandwidth for the farmer to load the acquired footage videos as if any intrusion issues happen.

4.3 Requirement Analysis

The purpose of requirements analysis is to identify and analyze any tasks or requirements necessary to complete the project successfully. Data requirements, functional requirements, non-functional requirements, and other needs are all part of the content to be examined in this project. Hence, the two categories, the hardware and software requirements, will smoothly run the project functionality.

4.4 Data Requirement

The data required to produce the output from the input received and gathered is known as the data requirement. The input is triggered when the PIR sensor that is being used detects suspicious motions in the environment. It will respond by emitting an alarm sound through the buzzer as long as the movements continue. It indicates that intruders or even wild animals have entered the region. When the sensor is triggered, the user will receive messages informing them of the current status of their farm. As a result, because the user is not at the farm, this Smart Light Control does not cause them to be concerned about their farm's state.

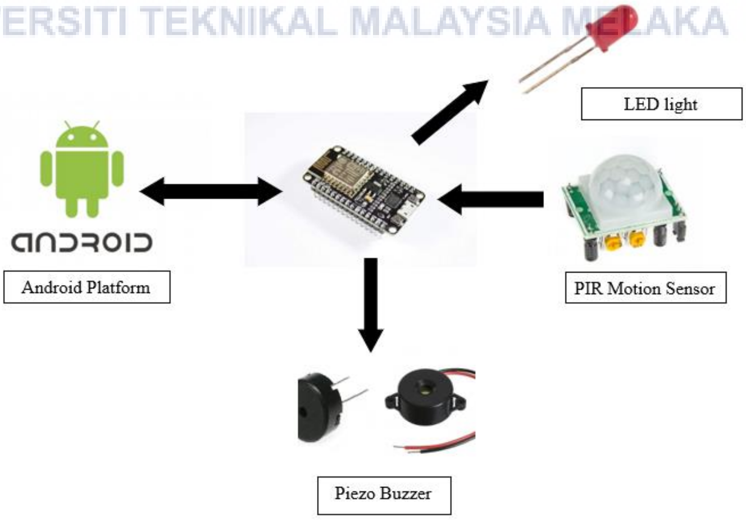


Figure 4.1: Data requirement flow

4.5 Functional Requirement

This functional requirement section will specify the projects based on the specific functions. This section will get through each function in further detail to make it more relevant systematic and understandable. When the functional need is fulfilled with specifications, this project will be able to operate and function smoothly. As a result, this section will discuss notification and motion detection.

4.5.1 Notification Process

For this notification process part, there will be NodeMCU ESP8266 with the WiFi features that will connect the android apps. The user will be notified through the notifications alert as it detects motions. The sensor will be also being programmed through the combination with NodeMCU ESP8266 to ensure that it functions very well. Based on Figure 4.2 below shows the notification process of the project.

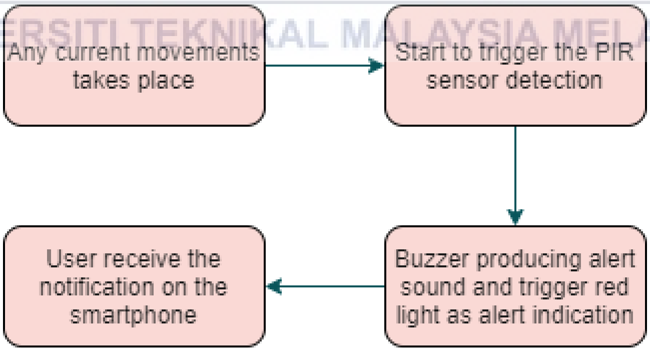

Any current movements takes place
Start to trigger the PIR sensor detection
Buzzer producing alert sound and trigger red light as alert indication
User receive the notification on the smartphone

Figure 4.2: Notification process

4.5.2 Motion Detection Process

Next, for the motion detection process, the PIR sensor plays the main role here. After any suspicious motion triggers it, the buzzer will start the ring as the sign to indicate an intrusion is happening. Hence, it will not stop ringing as long as the motions are still mingling around the PIR sensor motion detection range. Figure 4.3 shows the flow of the motion detection process.

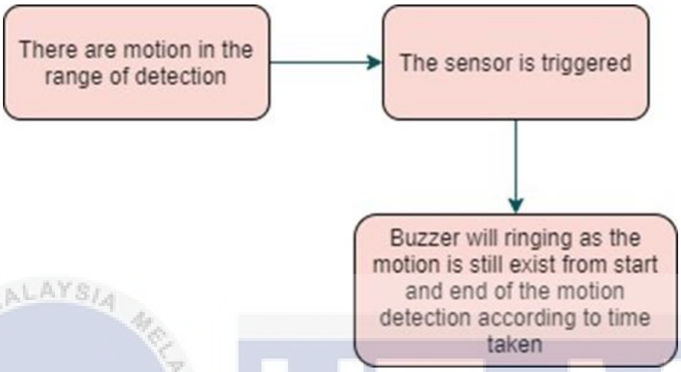

There are motion in the range of detection
The sensor is triggered
Buzzer will ringing as the motion is still exist from start and end of the motion detection according to time taken

Figure 4.3: Motion detection process

4.5.3 Light Control Process

Next, the user can control whether to be on or off via the mobile application interface for the light control process. They can also control the brightness level of the lights according to the brightness level that they want. Figure 4.4 shows the flow for the motion detection process.

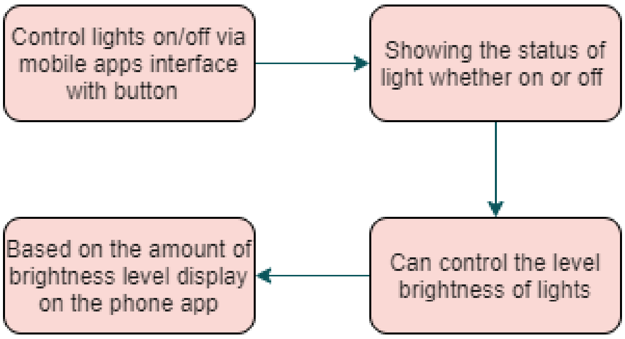

Control lights on/off via mobile apps interface with button
Showing the status of light whether on or off
Can control the level brightness of lights
Based on the amount of brightness level display on the phone app

Figure 4.4: Light Control process

4.6 Non-Functional Requirement

It is unrelated to non-functional requirements with those that are not directly related to the project's specific utility. In simple terms, a non-functional requirement depicts the framework's activity capacities and imperatives that improve its usefulness.

4.7 Other Requirement

The other requirements for other essentials are based on product and equipment requirements. Hence, the product and equipment requirements are the specifications for all items and equipment employed in the project to create a framework.

4.7.1 Hardware Requirement

1. Node MCU ESP8266

   The Node MCU ESP8266 is the main component of this project. It acts as the microcontroller, including the Wi-Fi functionality as one of the Node MCU development board features. This board contains a low-cost chip called ESP8266 that uses the TCP/IP protocol. It also consists few GPIO pin slots to connect the other hardware component to its body. Hence, this Node MCU version 2 in black was chosen for this project.

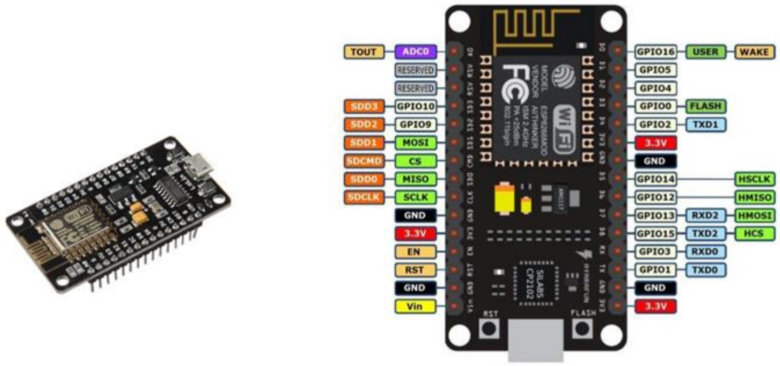
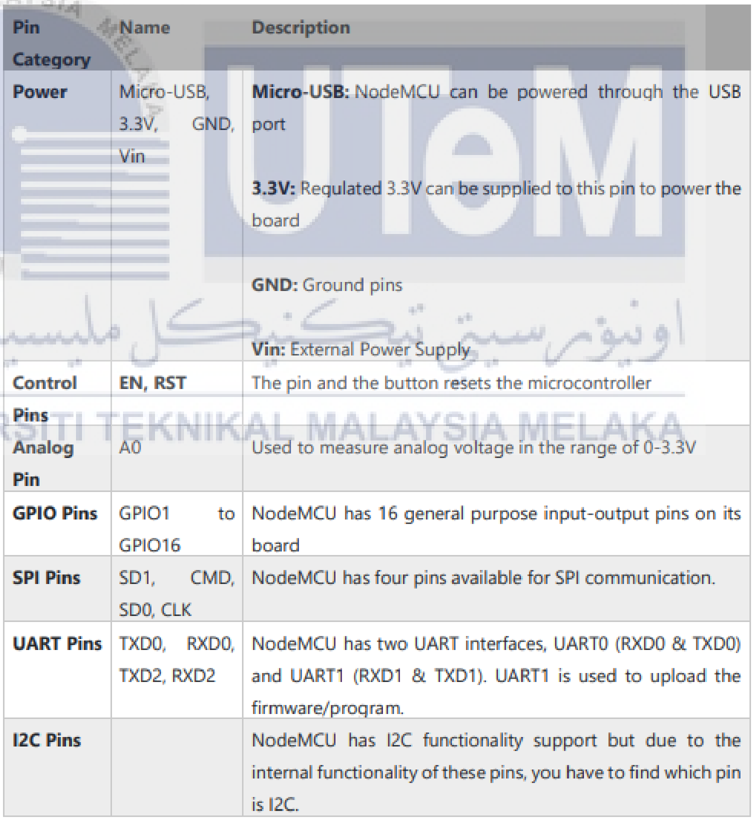


Figure 4.5: Node MCUESP8266 module



Pin Category	Name	Description
Power	Micro-USB, 3.3V, GND, Vin	**Micro-USB:** NodeMCU can be powered through the USB port **3.3V:** Regulated 3.3V can be supplied to this pin to power the board **GND:** Ground pins **Vin:** External Power Supply
Control Pins	**EN, RST**	The pin and the button resets the microcontroller
Analog Pin	A0	Used to measure analog voltage in the range of 0-3.3V
GPIO Pins	GPIO1 to GPIO16	NodeMCU has 16 general purpose input-output pins on its board
SPI Pins	SD1, CMD, SD0, CLK	NodeMCU has four pins available for SPI communication.
UART Pins	TXD0, RXD0, TXD2, RXD2	NodeMCU has two UART interfaces, UART0 (RXD0 & TXD0) and UART1 (RXD1 & TXD1). UART1 is used to upload the firmware/program.
I2C Pins		NodeMCU has I2C functionality support but due to the internal functionality of these pins, you have to find which pin is I2C.

Table 4.1 NodeMCU ESP8266 specification details

2. USB Cable A

The cable-A adapter will be used to link the Arduino IDE and the Node MCU ESP8266 in this project. As a result, the user will be able to establish the connection between the hardware Arduino board and the software via the programmed code.

Figure 4.6: USB cable A type

3. PIR sensor

The PIR Sensor stands for the passive infrared sensor. It is used in applications that require the detection of human or particle movement within a specific range. Hence, this sensor detects the movement of items that emit infrared light (such as human bodies). The output of the PIR motion detection sensor can be directly linked to one of the digital pins of an Arduino (or any other microcontroller).

Figure 4.7: PIR sensor

4. Jumper Wire

The jumper wires are simply wires having connector pins on both ends that can be used to connect two places without the use of solder. Jumpers are frequently used with breadboards and other prototyping equipment to allow for quick circuit replacement. Male to male, male to female, and female to female are the three most common forms of jumpers. Males are the most frequently used users.

Figure 4.8: Jumper wire

5. Light Emitting Diode (LED)

A light-emitting diode (LED) is a two-lead semiconductor light source. The LED emits light when it is activated. In this project, the LED is activated when the motion is being detected. Red LED is being chosen to indicate an alert through the light that sparks from the LED. Figure 4.10 shows the example of LED used in this project

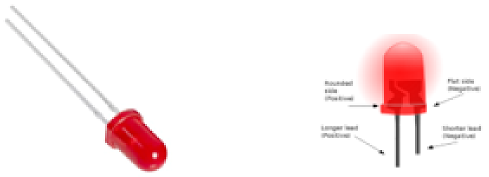


Figure 4.9: LED component

6. Piezo Buzzer

The buzzer shown below is known as the Piezo buzzer. Hence it is a component that was employed in the development of the project's prototype. The buzzer then emits a tone of 1000. The loop () procedure will repeat this process, making a short beeping noise each time or as long as it detects the motion. The buzzer hardware is shown in Figure 4.11.

Figure 4.10: Piezo Buzzer

7. Resistor 220 K

Next, in the electronic circuit, a resistor is the electrical component that limits or governs the flow of electrical current. Resistors are then used to, among other things, reduce current flow, regulate signal levels, divide voltages, bias active devices, and terminate transmission lines. Figure 4.12 depicts an example of a resistor that was utilized in this project.

Figure 4.11: 220K ohm resistor

8. Breadboard

The Breadboard is a great way to connect other tools with Arduino UNO through its pin. These are for making prototyping the smart Light Control through the circuit. The breadboard will be used to connect with the GPIO pin at the Arduino board to allow the LED, buzzer, resistor and others to function properly. Figure 4.13 shows the example of breadboard used in this project

Figure 4.12: Breadboard

9. Android Smartphone

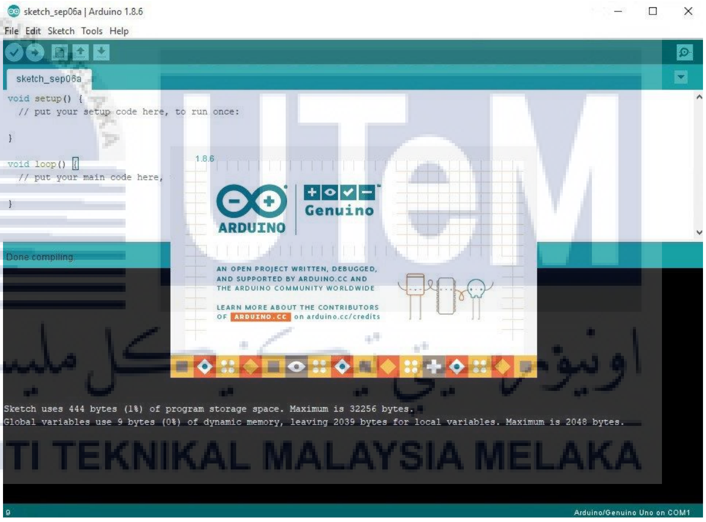
The Android Smartphone is one of the important hardware that needs to be used in testing the project. So, for the project testing, a Huawei Nova 2i is being used with a RAM capacity of 4GB and internal storage of 64 GB. Both EMUI and Android is 8.0.0 version. Figure 4.14 shows the Android smartphone used in this project.

Figure 4.13: Huawei Nova 2i

4.7.2 Software Requirement

1. Arduino IDE

The Arduino IDE is a multi-platform application that runs on Windows, Mac OS X, and Linux. The program writing and uploading software for Arduino compatible boards. This is how the Java programming language communicates. It comes with the GNU public license version 2, which was just released on the market. The restructure is done using the Arduino IDE's language C and C ++ code.


sketch_sep06a | Arduino 1.8.6
File Edit Sketch Tools Help
sketch_sep06a
1.8.6
ARDUINO
Genuino
Arduino/Genuino Uno on COM1

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here,

}
```

Done compiling.

AN OPEN PROJECT WRITTEN, DEBUGGED, AND SUPPORTED BY ARDUINO.CC AND THE ARDUINO COMMUNITY WORLDWIDE

LEARN MORE ABOUT THE CONTRIBUTORS OF ARDUINO.CC on arduino.cc/credits

```
Sketch uses 444 bytes (1%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.
```

Figure 4.14: Arduino IDE Software

2. MIT App Inventor

The MIT Inventor Apps is being chosen because it is more precise programming and more straightforward to integrate. MIT Inventor App will be used as the project's operating system. Furthermore, it is simple to create completely complete applications. The MIT App is available for Android and iOS. It is open-source software that is free to use. As we all know, an operating system is a collection of fundamental programs and utilities that run on a specified schedule.

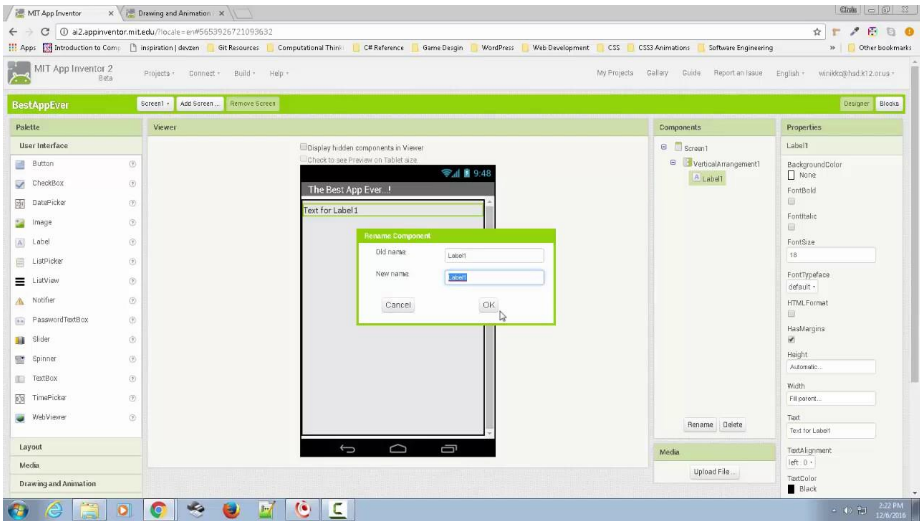


Figure 4.15: MIT Apps Inventor

3. Fritzing

Fritzing is an online open-source software tool for creating 3D models by beginners. Fritzing is a simple and easy-to-use online 3D modeling program that runs in a web browser and is available for download. As a result, it is a simple software for 3D design, electronics, and coding.

Figure 4.16: Fritzing Software

4. ThingSpeak

Tinker cad is an online platform known as a cloud-based IoT analytics tool that can aggregate, visualize, and analyze live data streams. It also allows the delivery of real-time visualizations of data sent to ThingSpeak by user's devices.

Figure 4.17: ThingSpeak Logo

4.8 High-Level Design

The structure of this project with the employment of IoT technology will also be described in the high-level design. It will go through the agent architecture, system design, and system functions in greater depth. Even for end-users unfamiliar with technical reports in information technology, high-level design is simple to understand.

4.8.1 System Architecture

The system architecture for this project is shown in the figure below. The technological framework, end-user requirements, and a list of system components such as hardware and software must all be included in system architecture. The user will use an Android smartphone as the user interface with the integration of the application that can be used to control or manage the light at their small-scale farm.

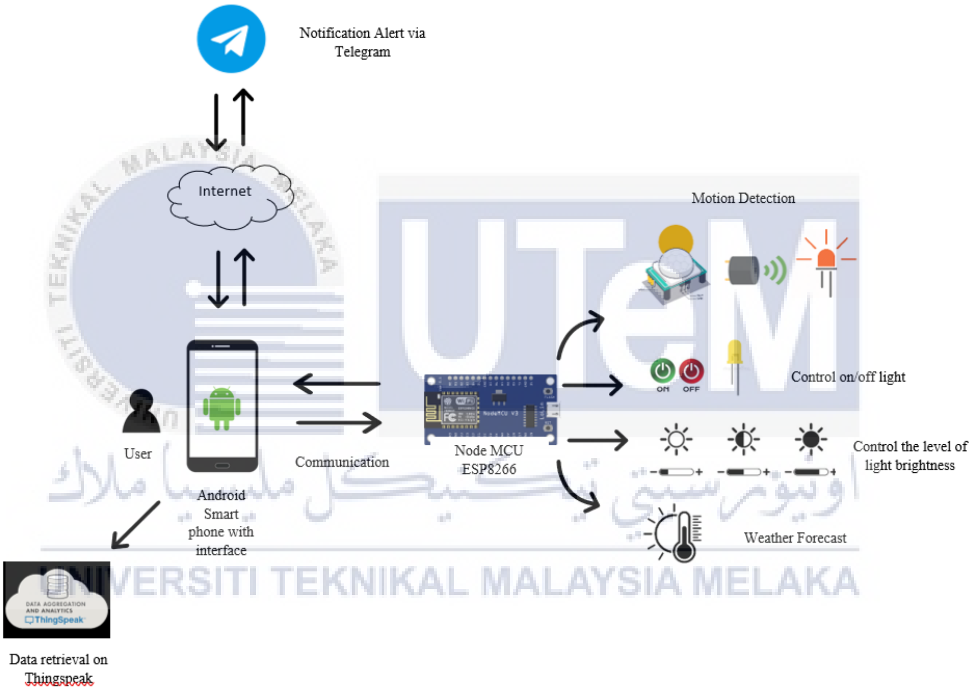

Notification Alert via Telegram
Internet
Motion Detection
User
Android Smart phone with interface
Communication
Node MCU ESP8266
ON
OFF
Control on/off light
Control the level of light brightness
Weather Forecast
DATA AGGREGATION AND ANALYTICS
ThingSpeak
Data retrieval on Thingspeak

Figure 4.18: System Architecture Diagram

4.8.2 User Interface Design

For this project, the graphical user interface (GUI) will be created in order to make it more effective and efficient. This system's user interface must be precise and straightforward so that users can grasp it. The user-friendly is also one factor that needs to be looked up for the user to flow the app and understand the use of the button created on the interface.

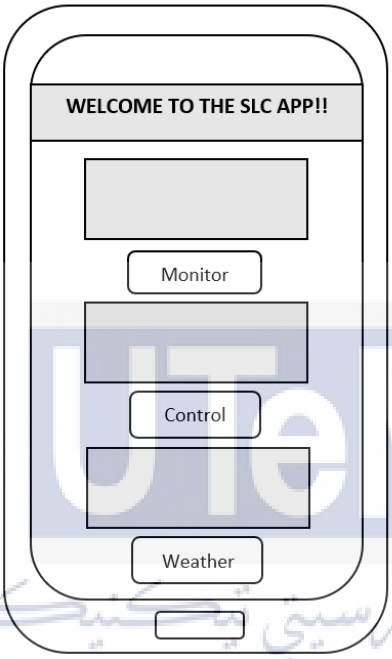
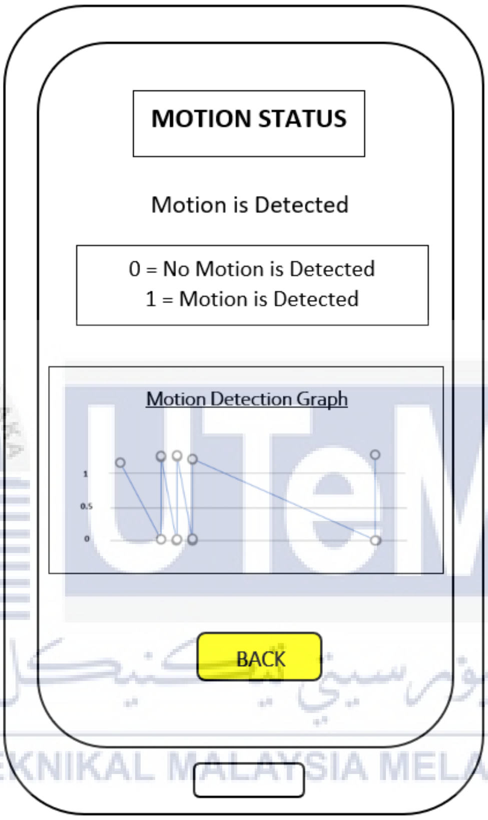


Figure 4.19: First Interface

Figure 4.20 shows the main interface of the Smart Light Control application. The interface is being designed in a very simple and modest way to make sure that the users could understand the use of the button display in the picture. Such as the monitor or control button, so if they insist on inspecting the farm condition, they can hit the monitor and otherwise for the control button. There is also a current timestamp available, including the date and day.

Next, Figure 4.21 shows the second interface, the Motion Detection interface, where it will be navigated after the user chooses to press the monitor button. This interface will display the current status of the farm motion detection status and the motion's real-time graph.



Motion is Detected

0 = No Motion is Detected
1 = Motion is Detected

Figure 4.20: Second Interface

Figure 4.22 shows the third interface which will be navigated after the user chooses to press the control button. This interface will have few functionalities where users can control the lights via the Android smartphone whether to turn on or turn off the light.

The light status also would be display on the interface showing the light condition. In addition, the user can also control the brightness level of the lights according to how bright they want. It will also display the amount of brightness on the screen.

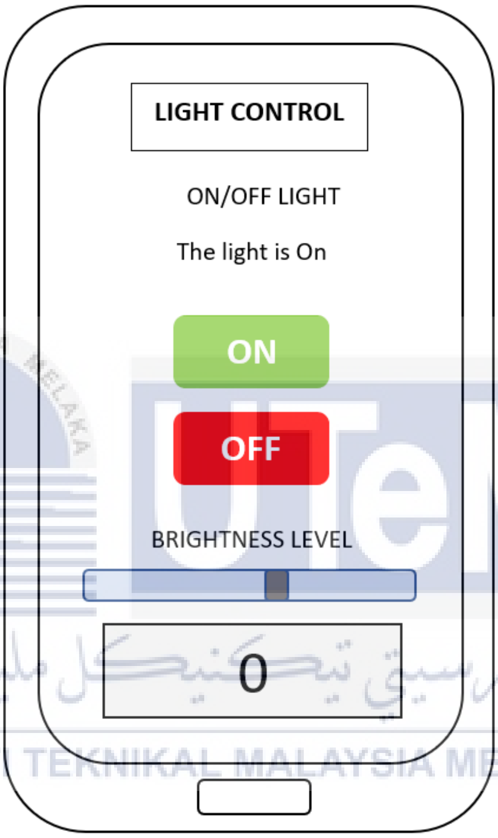


Figure 4.21: Third Interface

Figure 4.23 shows the fourth interface design, which is the weather forecast information view. This interface is one of the additional functionalities of the application where the user can view the current weather of the day by just viewing the real-time data according to their location. The user just has to press the get weather button and it will display the weather.

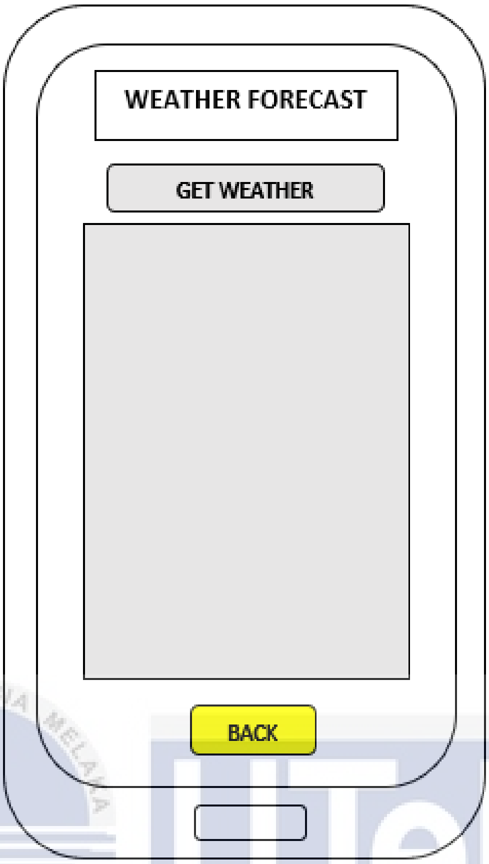


Figure 4.22: Fourth Interface

4.9 Flowchart Design

A flowchart is also included in the system architecture. In flowchart design, it depicts the input, output, data processing, and decision-making. Each step is represented graphically or symbolically in a flowchart. Each stage in the process is represented by a separate symbol, which includes a brief description of the stage as well as an arrow representing the process flow. The workflow or process of the smart light control with the IoT technology is represented by the system flow chart design

4.9.1 Smart Light Control Application Flowchart

The figure below is a brief view of the flow in the smart light control application. In fact, with the aid of this flowchart, it helps or facilitates understanding the process of the system itself with the use of flowchart symbols in a sequential order such as the oval shape that indicates the starting point of the flow. This rectangle indicates the process that happens at that stage, moving to the diamond shape, which indicates the decision that will decide the next subprocess.

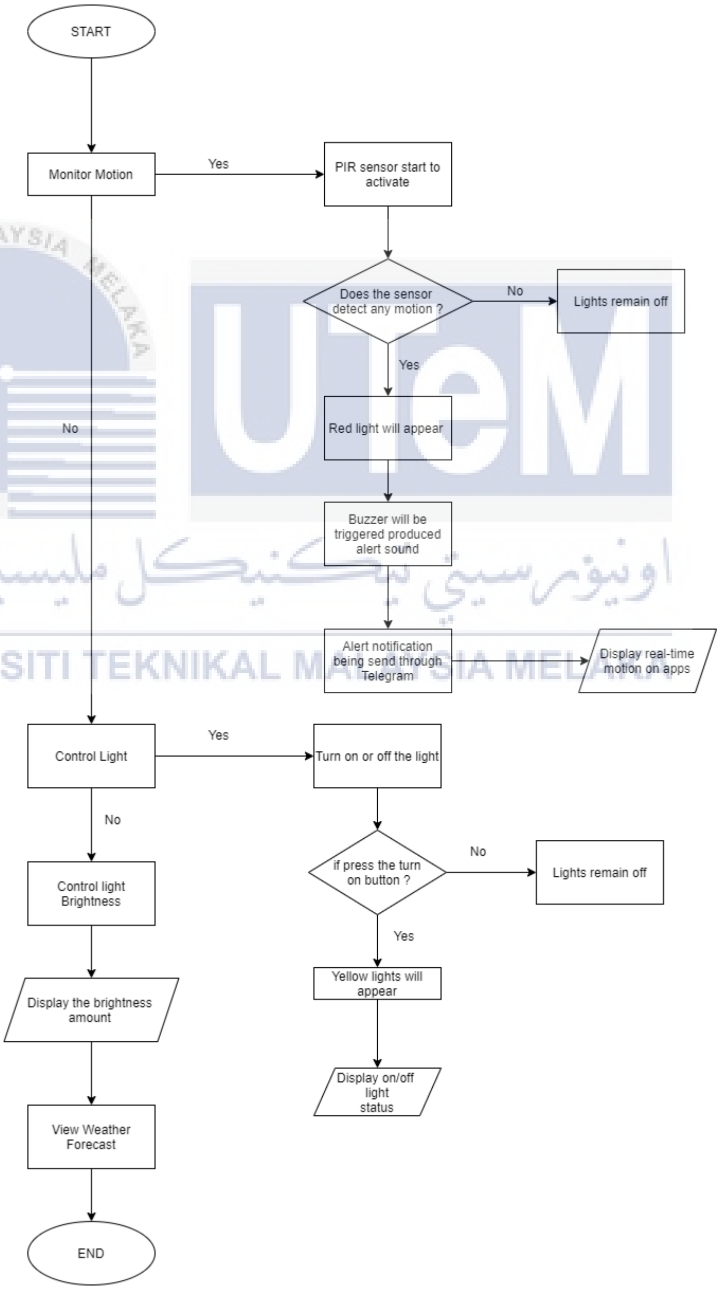

START
Monitor Motion
Yes
PIR sensor start to activate
Does the sensor detect any motion ?
No
Lights remain off
Yes
Red light will appear
No
Buzzer will be triggered produced alert sound
Alert notification being send through Telegram
Display real-time motion on apps
Control Light
Yes
Turn on or off the light
No
if press the turn on button ?
No
Lights remain off
Control light Brightness
Yes
Yellow lights will appear
Display the brightness amount
Display on/off light status
View Weather Forecast
END

Figure 4.23: System Flowchart

4.9.2 System Implementation Flowchart

Based on the figure shown below is a quick view of the flow in the system implementation. This shows the interrelation between the NodeMCU ESP 8266 that act as a microcontroller with the Wi-Fi module embedded on its board to establish connection between the hardware and the software part. So, the flowchart below shows the hardware component flow functionality process.

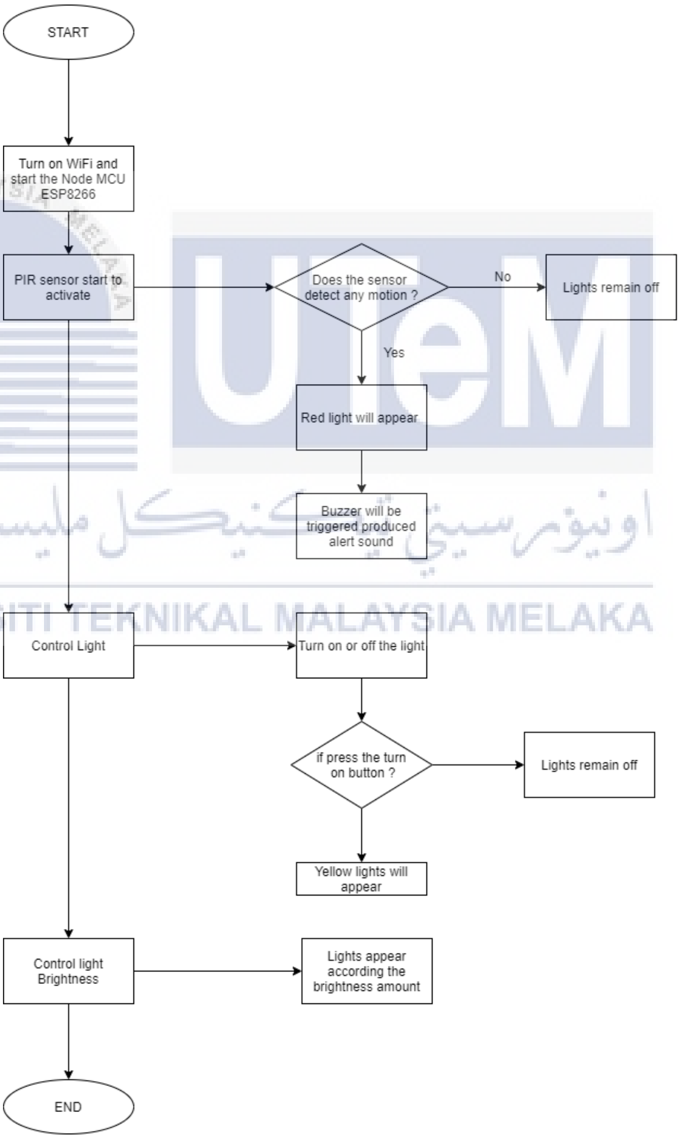

START
Turn on WiFi and start the Node MCU ESP8266
PIR sensor start to activate
Does the sensor detect any motion ?
No
Lights remain off
Yes
Red light will appear
Buzzer will be triggered produced alert sound
Control Light
Turn on or off the light
if press the turn on button ?
Lights remain off
Yellow lights will appear
Control light Brightness
Lights appear according the brightness amount
END

Figure 4.24: System Implementation Flowchart

4.10 Conclusion

One of the most significant aspects of the realization project is the analysis and design of this chapter. All software and hardware requirements must be defined and researched before the project can be implemented. Hence, the execution phase can begin after the design has been accepted. This chapter covers the implementation preparation step as well as the overall system flow to increase the understanding of the project concept. The implementation phase and projected output from will be discussed in the next following chapter.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

This chapter explains how the Smart Light Control with Notification Alert for Small-scale farm projects can be implemented in software and hardware development. The design results are communicated to the logical procedure and software code during the system application stage. During this phase, the system will be tested to ensure that the design meets the required standards. The implementation stage involves developing and testing the functional scheme that fits the requirements for organizational design and implementing the system-to-present system interface. It necessitates the creation of a database, an application, and user and system interfaces.

5.2 Development Environment Setup

The hardware and software specs would be included in the design of the project system's construction process. All alterations will be detailed in detail and visible. The hardware and software specs are put forth in Chapter 4. They will be further explained in the association's chapter following where all the modifications will be described step by step and seen clearly.

5.2.1 Hardware Development Environment Setup

Chapter 4 describes the hardware utilization in this project. Node MCU ESP8266, PIR Sensor, Piezo Buzzer, resistor, LED and jumper wire are all included. The PIR sensor will be connected to the Node MCU ESP8266 board. All of these hardware components will be connected via the jumper wire. The motions will be detected by the PIR motion sensor. It can detect motion with its specified radius range and wave frequency of the human body or any living things that emitted heat.

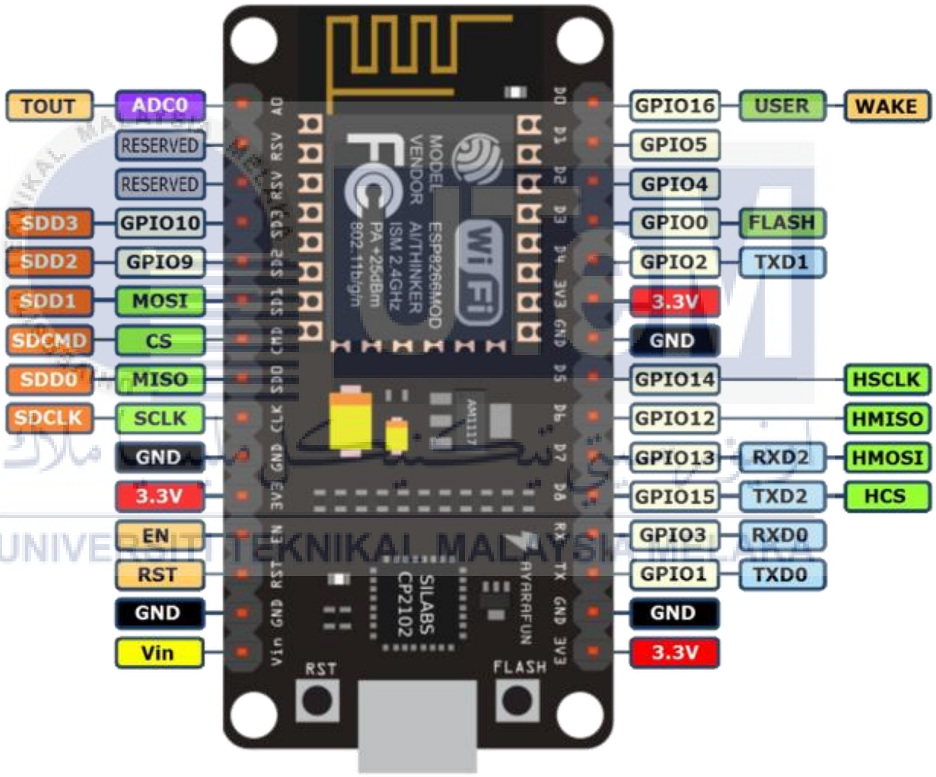


Figure 5.1: Details of the Node MCU ESP8266 board

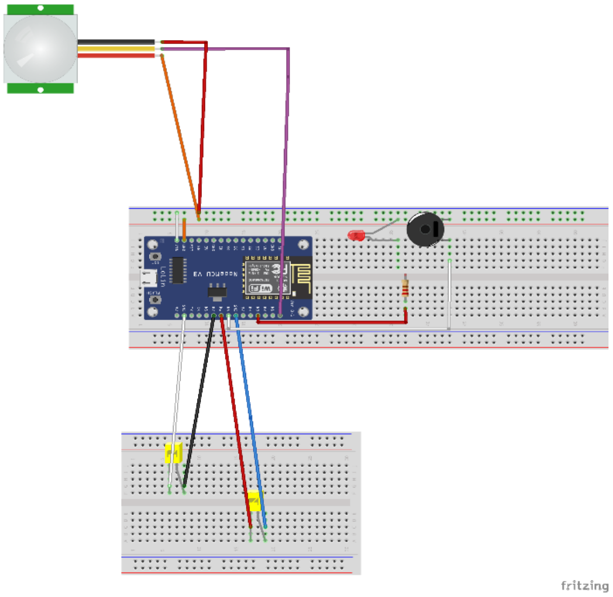


Figure 5.2: Hardware Diagram

Figure 5.2 shows a diagram sketch of the microcontroller's integration and other components part of the hardware utilization in this project. Meanwhile in figure 5.3 shows the hardware development setup.

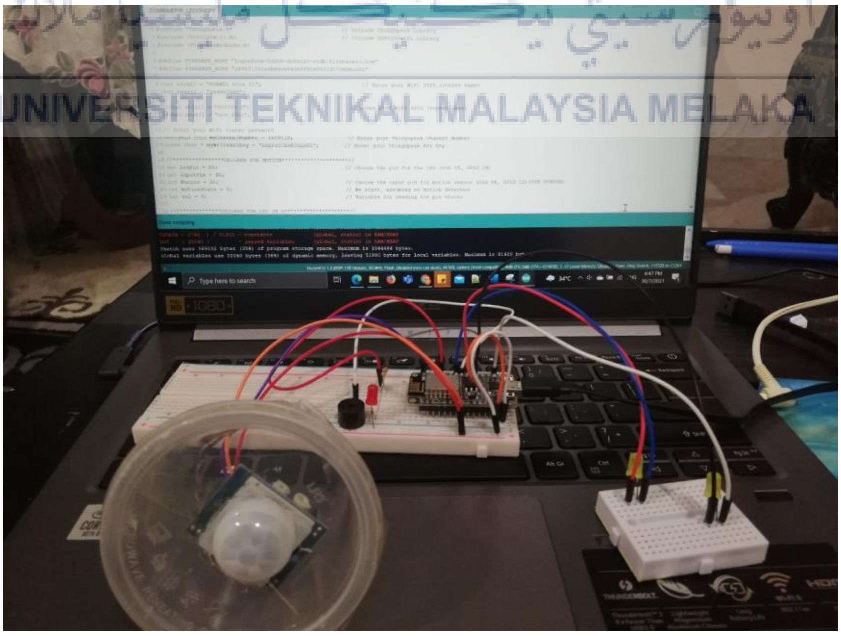


Figure 5.3: Hardware development prototype

5.2.2 Software Development Environment Setup

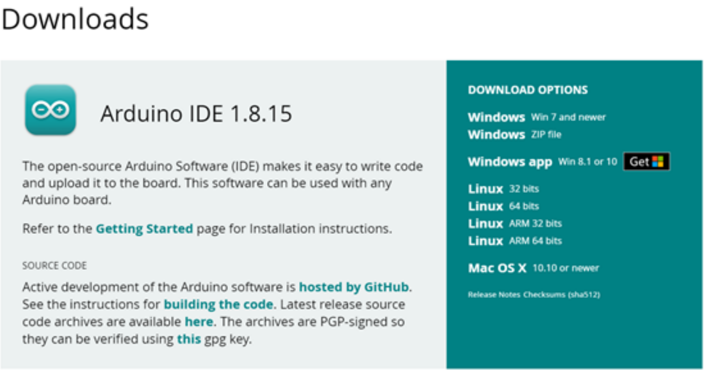
This sub-topic will discuss in-depth the project development software process. Furthermore, this project necessitates the use of both hardware and software in order for the project to function properly and entirely. Hence, all of these components will facilitate the user to participate more in the project's development.

1. Arduino Integrated Development Environment (IDE) setup

Next, the Arduino IDE was used to configure the code and upload the source code to the microcontroller in this project. The Arduino IDE is a software development environment that enables the hardware to function.

When the Arduino IDE receives a motion sensor value, it can respond with the alert sound buzzer and red lights on from the LED. Hence, the Arduino IDE sends the receiving command from the Wi-Fi module through the serial port.

Step 1: Firstly, is download and install the Arduino IDE platform on the Arduino official website at https://www.arduino.cc/en/software.



Downloads

Arduino IDE 1.8.15

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the **Getting Started** page for Installation instructions.

SOURCE CODE

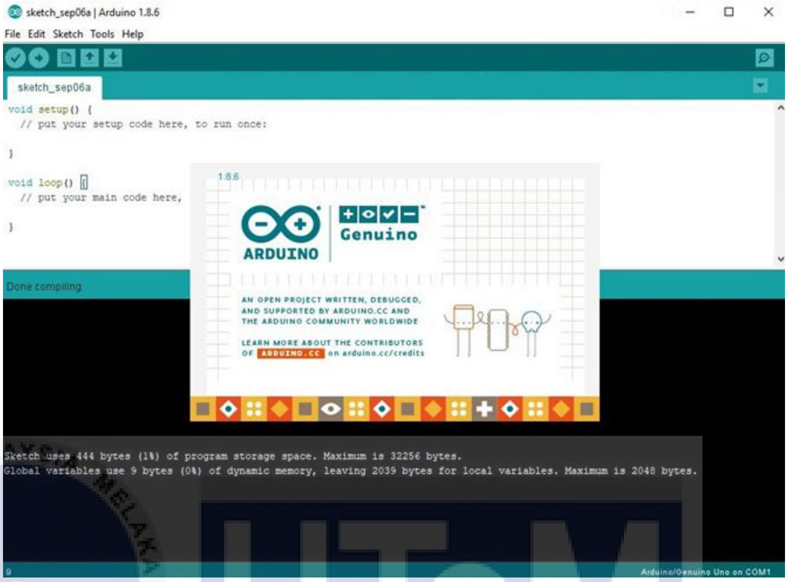
Active development of the Arduino software is **hosted by GitHub**. See the instructions for **building the code**. Latest release source code archives are available **here**. The archives are PGP-signed so they can be verified using **this** gpg key.

DOWNLOAD OPTIONS

Windows Win 7 and newer
Windows ZIP file
Windows app Win 8.1 or 10 Get
Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits
Mac OS X 10.10 or newer
Release Notes Checksums (sha512)

Figure 5.4: Download Arduino IDE

Step 2: Next, run the Arduino IDE software and wait for a while until the initialization process is done.


sketch_sep06a | Arduino 1.8.6
File Edit Sketch Tools Help
sketch_sep06a
1.8.6
ARDUINO
Genuino
AN OPEN PROJECT WRITTEN, DEBUGGED, AND SUPPORTED BY ARDUINO.CC AND THE ARDUINO COMMUNITY WORLDWIDE
LEARN MORE ABOUT THE CONTRIBUTORS OF ARDUINO.CC on arduino.cc/credits

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here,

}
```

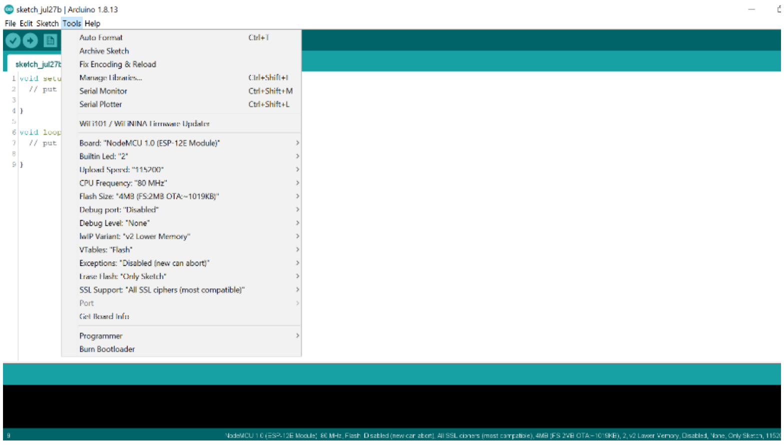
Done compiling.

```
Sketch uses 444 bytes (1%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.
```

Arduino/Genuino Uno on COM1

Figure 5.5: IDE packages initialization

Step 3: The first interface will display the page where the settings, code modification and implementation are being done. Then, do some setup by choosing the correct microcontroller and port to be configured.

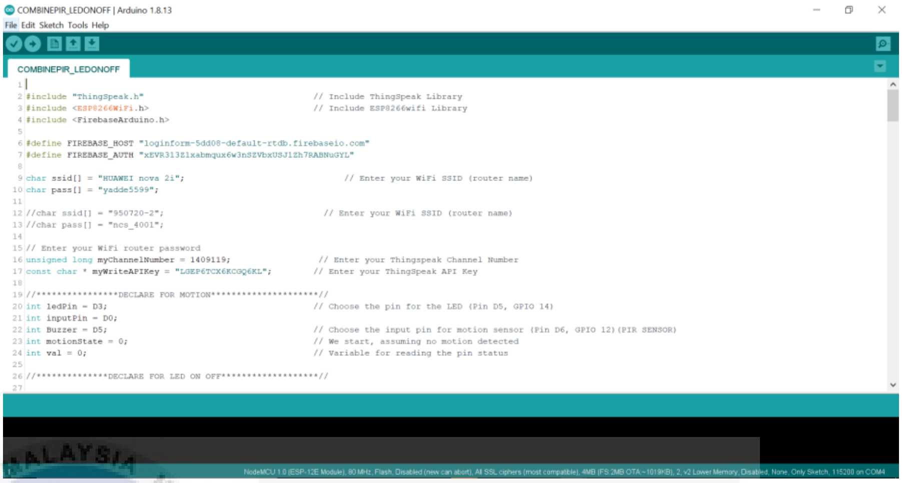

sketch_jul27b | Arduino 1.8.13
File Edit Sketch Tools Help
sketch_jul27b
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Port
Get Board Info
Programmer
Burn Bootloader

Board: "NodeMCU 1.0 (ESP-12E Module)"
Builtin Led: "2"
Upload Speed: "115200"
CPU Frequency: "80 MHz"
Flash Size: "4MB (FS:2MB OTA:~1019KB)"
Debug port: "Disabled"
Debug Level: "None"
lwIP Variant: "v2 Lower Memory"
VTables: "Flash"
Exceptions: "Disabled (new can abort)"
Erase Flash: "Only Sketch"
SSL Support: "All SSL ciphers (most compatible)"

NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled (new can abort), All SSL ciphers (most compatible), 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 1152

Figure 5.6: Arduino IDE interface

Step 4: Configure the functionality with the respective source code and upload it into the microcontroller, the Node MCU ESP8266.


COMBINEPIR_LEDONOFF | Arduino 1.8.13
File Edit Sketch Tools Help
COMBINEPIR_LEDONOFF

```

#include "ThingSpeak.h"                                // Include ThingSpeak Library
#include <ESP8266WiFi.h>                               // Include ESP8266wifi Library
#include <FirebaseArduino.h>

#define FIREBASE_HOST "loginform-5dd08-default-rtdb.firebaseio.com"
#define FIREBASE_AUTH "xEVR313Z1xabmqux6w3nSZVbxUSJ1Zh7RABNuGYL"

char ssid[] = "HUAWEI nova 2i";                              // Enter your WiFi SSID (router name)
char pass[] = "yadde5599";

//char ssid[] = "950720-2";                                // Enter your WiFi SSID (router name)
//char pass[] = "ncs_4001";

// Enter your WiFi router password
unsigned long myChannelNumber = 1409119;                // Enter your Thingspeak Channel Number
const char * myWriteAPIKey = "LGEP6TCX6KCGQ6KL";       // Enter your ThingSpeak API Key

//****************DECLARE FOR MOTION*********************//
int ledPin = D3;                                       // Choose the pin for the LED (Pin D5, GPIO 14)
int inputPin = D0;
int Buzzer = D5;                                       // Choose the input pin for motion sensor (Pin D6, GPIO 12)(PIR SENSOR)
int motionState = 0;                                   // We start, assuming no motion detected
int val = 0;                                           // Variable for reading the pin status

//**************DECLARE FOR LED ON OFF*******************//

```

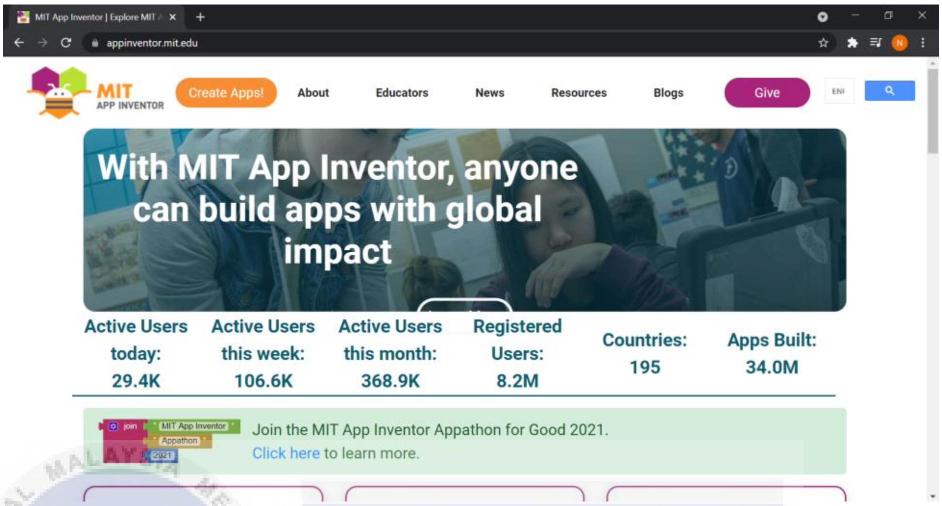
NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled (new can abort), All SSL ciphers (most compatible), 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM4

Figure 5.7: Configure and upload source code on IDE

2. MIT APP Inventor Setup

The MIT App Inventor is one of the online open-source platforms designed for any Android mobile applications development purpose. Hence, with the integration of some features by just simply dragging and dropping any desired components in the workspace to produce a mobile view interface. Meanwhile, in terms of its functionality, the part can be done via constructing the block code.

Step 1: Open the MIT Apps Inventor platform on the browser and click on the "Create Apps" button to start developing the project.



With MIT App Inventor, anyone can build apps with global impact

Active Users today:	Active Users this week:	Active Users this month:	Registered Users:	Countries:	Apps Built:
29.4K	106.6K	368.9K	8.2M	195	34.0M

Join the MIT App Inventor Appathon for Good 2021.
Click here to learn more.

Figure 5.8: Main page MIT App Inventor

Step 2: Then, in order to start, choose the "Start new project button".

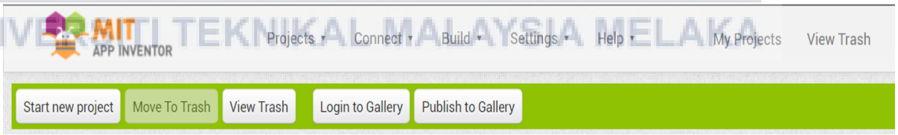
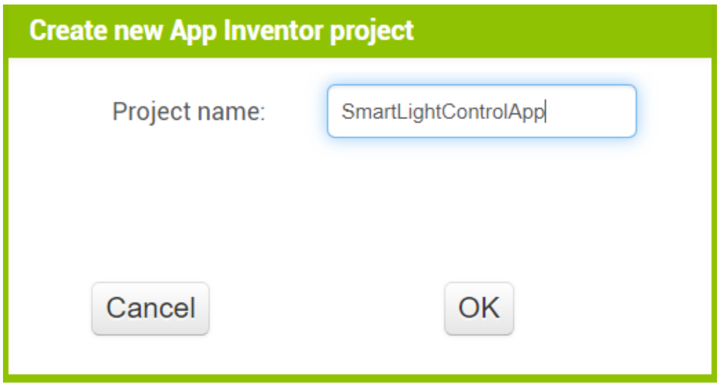


Figure 5.9: Create a new project

Step 3: Next, give a title name to the project created.



Create new App Inventor project

Project name: SmartLightControlApp

Figure 5.10: Application name

Step 4: Design the mobile application interface view

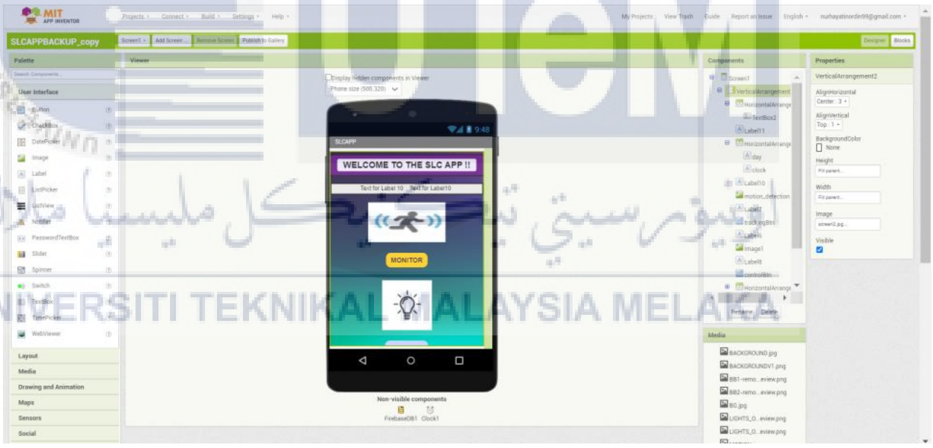


Figure 5.11: Design the mobile application

Step 5: For functionality, use the code block part to develop

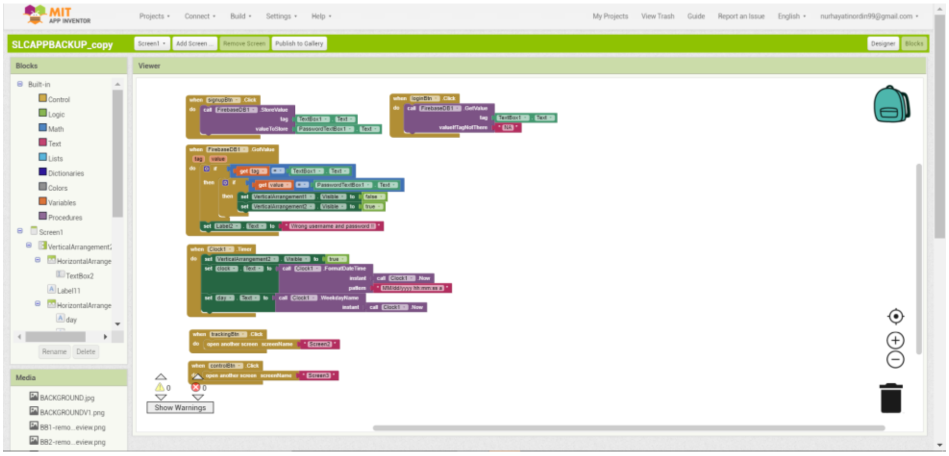
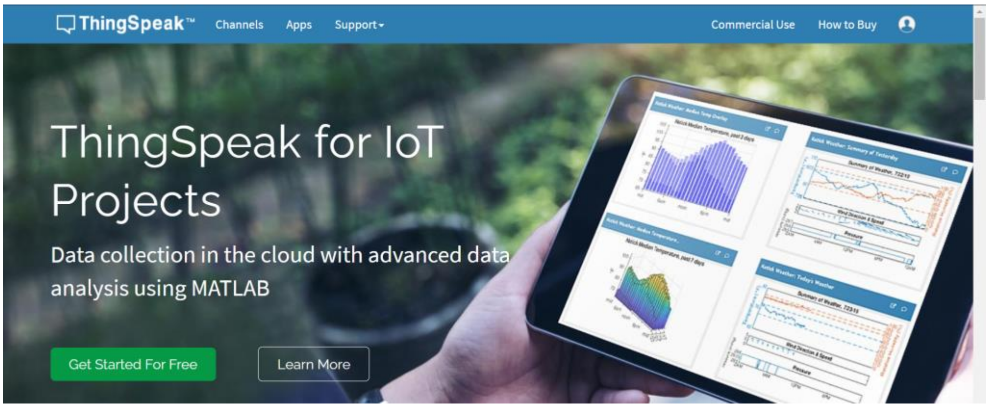


Figure 5.12: Code block development

3. ThingSpeak Setup

In this project, a cloud-based IoT analytics tool called ThingSpeak is being used as one of the mediums in collecting and gathering real-time data. It has the functionality of delivering real-time data representative.

Step 1: Open the ThingSpeak website and click the “Profile” button to start.


ThingSpeak for IoT Projects

Data collection in the cloud with advanced data analysis using MATLAB

Figure 5.13: ThingSpeak main website

Step 2: Click the "New Channel" button to create a new channel for the project.

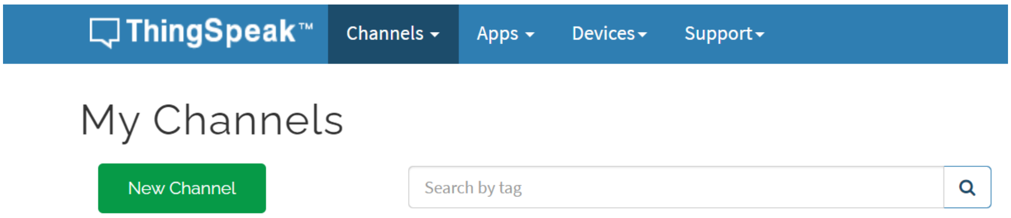
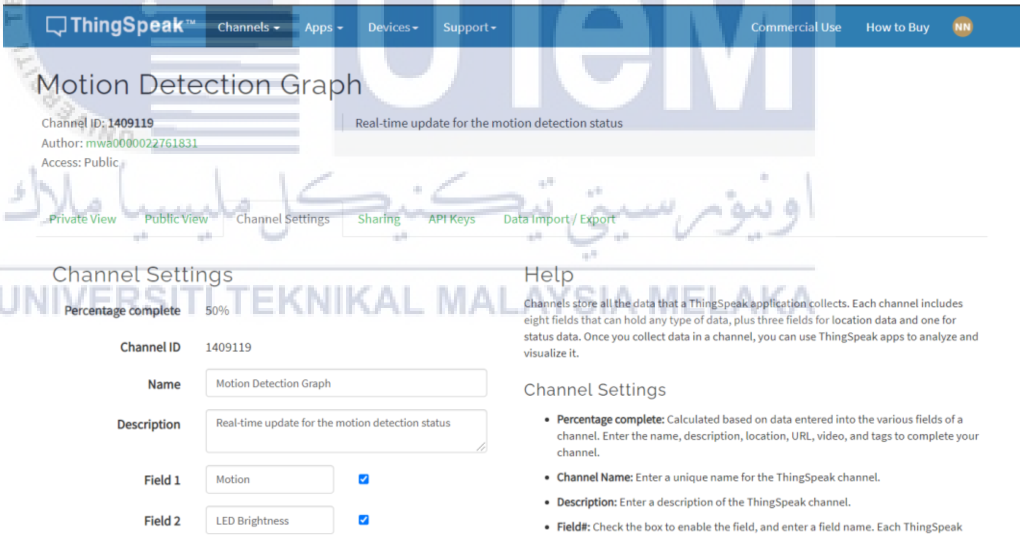


Figure 5.14: Create new channels for the project

Step 3: Then, fill in the field to construct the new channel to retrieve the real-time data obtained from the motion sensor.



Motion Detection Graph

Channel ID: **1409119**
Author: mwa0000022761831
Access: Public

Real-time update for the motion detection status

Channel Settings

Percentage complete 50%

Channel ID 1409119

Name Motion Detection Graph

Description Real-time update for the motion detection status

Field 1 Motion ☑

Field 2 LED Brightness ☑

Help

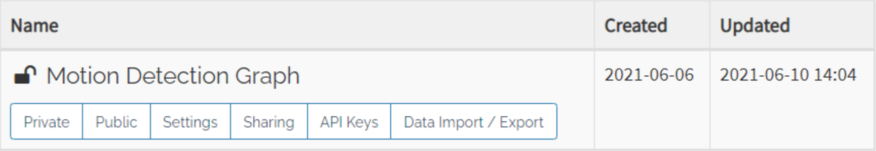
Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.

Channel Settings

- **Percentage complete:** Calculated based on data entered into the various fields of a channel. Enter the name, description, location, URL, video, and tags to complete your channel.
- **Channel Name:** Enter a unique name for the ThingSpeak channel.
- **Description:** Enter a description of the ThingSpeak channel.
- **Field#:** Check the box to enable the field, and enter a field name. Each ThingSpeak

Figure 5.15: ThingSpeak channels settings

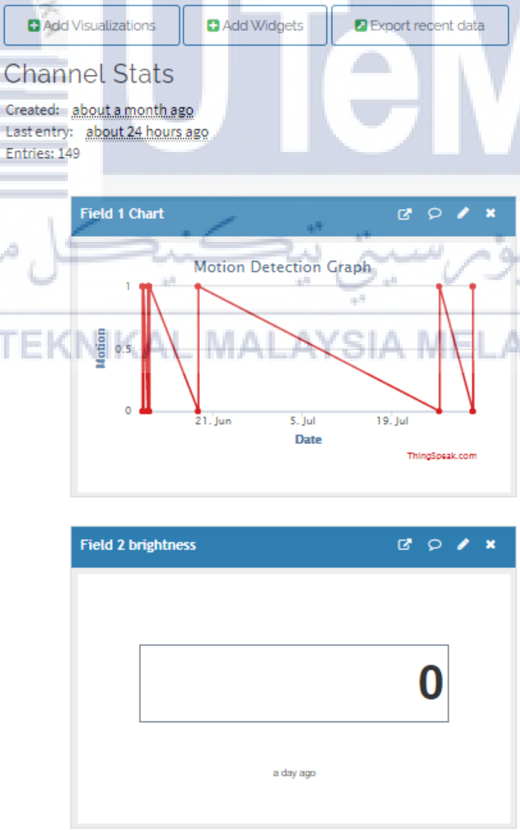
Step 4: Then, the ThingSpeak has been created according to the information that has been filled in previously.



Name	Created	Updated
Motion Detection Graph Private · Public · Settings · Sharing · API Keys · Data Import / Export	2021-06-06	2021-06-10 14:04

Figure 5.16: ThingSpeak created channel view

Step 5: Then, the data will be displayed on the ThingSpeak channel page for both motion sensor and the brightness control.



Channel Stats

Created: about a month ago

Last entry: about 24 hours ago

Entries: 149

Figure 5.17: The channel status

4. Telegram Bot Setup

Telegram Bot is one of the AI-inspired programs that are able to do multitask activities involving sending forecast weather updates, information, important news and many more. So, BotFather is being used in order to generate the respective bot for this project development.

Step 1: Firstly, on Telegram, search for " @BotFather ".

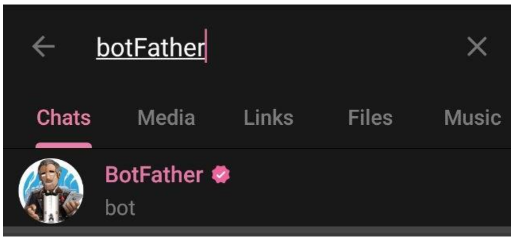
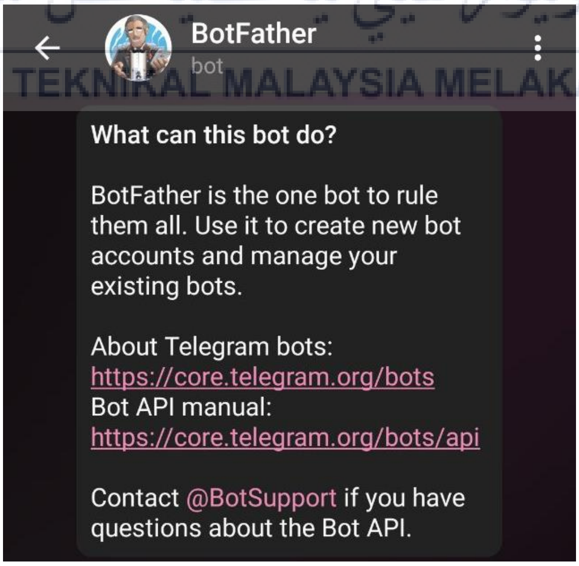


Figure 5.18: Search BotFather to generate a new bot

Step 2: Then, start by clicking on the “BotFather”.



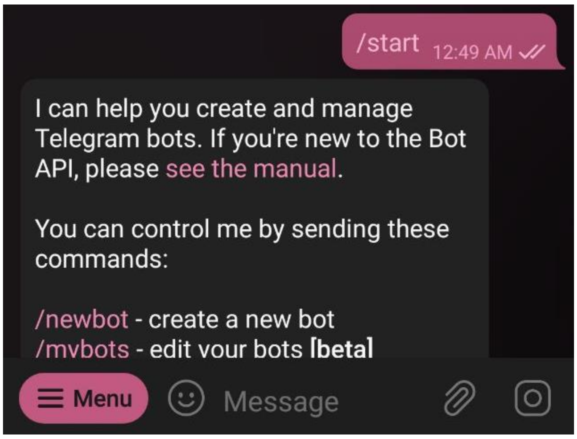
What can this bot do?

BotFather is the one bot to rule them all. Use it to create new bot accounts and manage your existing bots.

About Telegram bots:
https://core.telegram.org/bots
Bot API manual:
https://core.telegram.org/bots/api

Contact @BotSupport if you have questions about the Bot API.

Figure 5.19: Project bot development



/start 12:49 AM

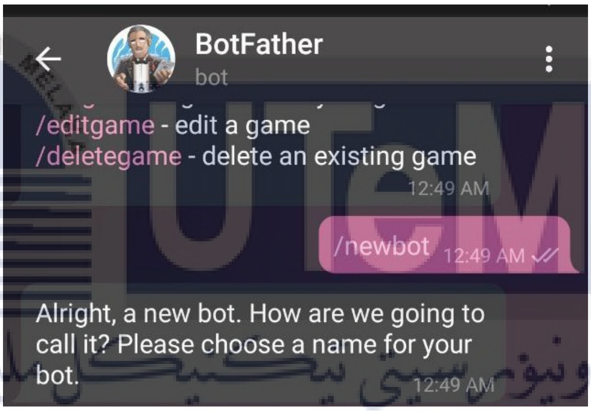
I can help you create and manage Telegram bots. If you're new to the Bot API, please see the manual.

You can control me by sending these commands:

/newbot - create a new bot
/mybots - edit your bots [beta]

Figure 5.20: Insert start command

Step 3: Next, create the new bot to be used.



BotFather
bot

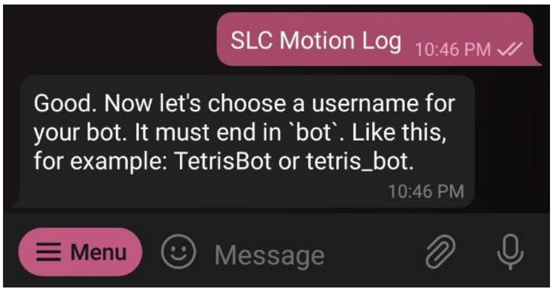
/editgame - edit a game
/deletegame - delete an existing game
12:49 AM

/newbot 12:49 AM

Alright, a new bot. How are we going to call it? Please choose a name for your bot. 12:49 AM

Figure 5.21: New bot command

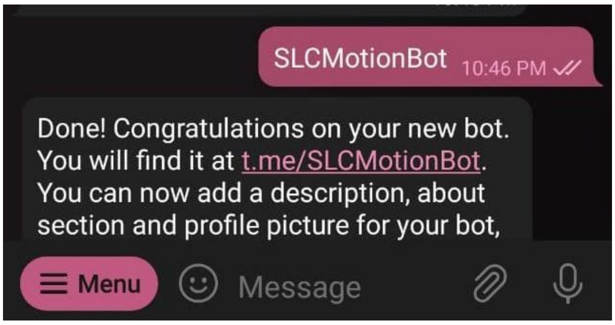
Step 4: Then, give a suitable name for the project Telegram bot.



SLC Motion Log 10:46 PM

Good. Now let's choose a username for your bot. It must end in `bot`. Like this, for example: TetrisBot or tetris_bot.
10:46 PM

Figure 5.22: Define name for the bot

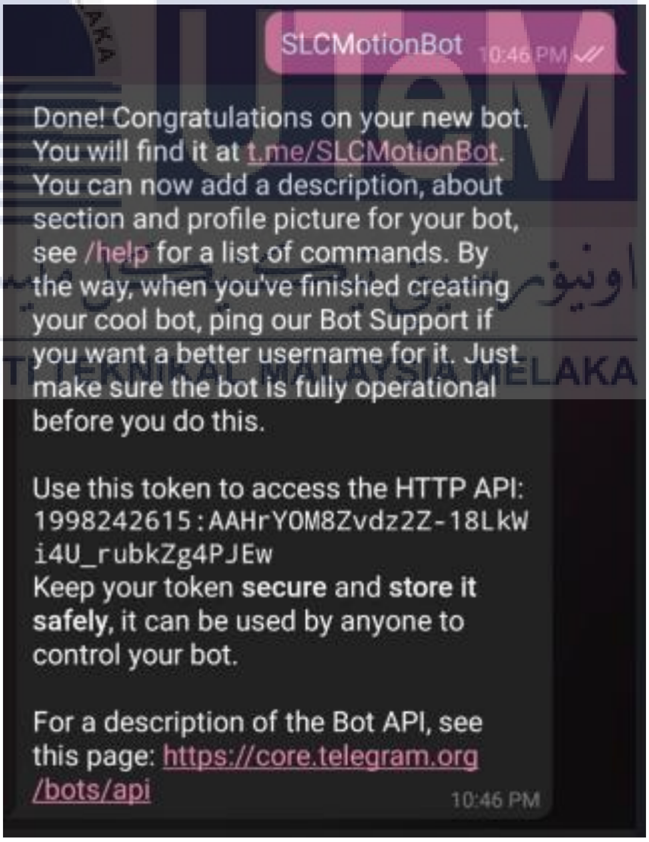


SLCMotionBot 10:46 PM

Done! Congratulations on your new bot. You will find it at t.me/SLCMotionBot. You can now add a description, about section and profile picture for your bot,

Figure 5.23: Create a username for the bot

Step 5: Then, successfully created the project's own bot.



SLCMotionBot 10:46 PM

Done! Congratulations on your new bot. You will find it at t.me/SLCMotionBot. You can now add a description, about section and profile picture for your bot, see /help for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.

Use this token to access the HTTP API:
1998242615:AAHrYOM8Zvdz2Z-18LkWi4U_rubkZg4PJEw
Keep your token **secure** and **store it safely**, it can be used by anyone to control your bot.

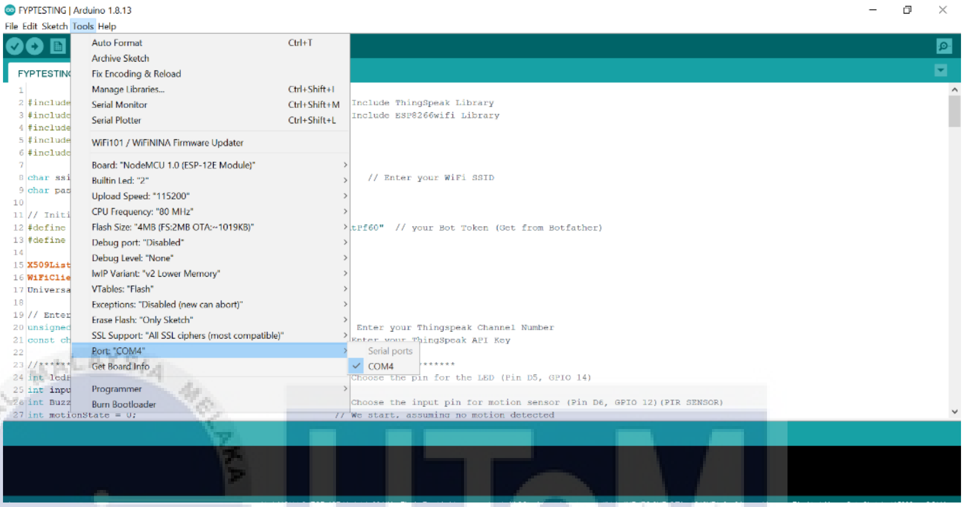
For a description of the Bot API, see this page: https://core.telegram.org/bots/api

10:46 PM

Figure 5.24: Bot successfully created

5.3 Arduino IDE Configuration Setup

1. Firstly, choose the NodeMCU 1.0 (ESP-12E Module) board microcontroller and select the port which is the port "COM4". This is to ensure that it will enable the mode and upload the code to the right source to establish the right connection between Arduino IDE and the device.

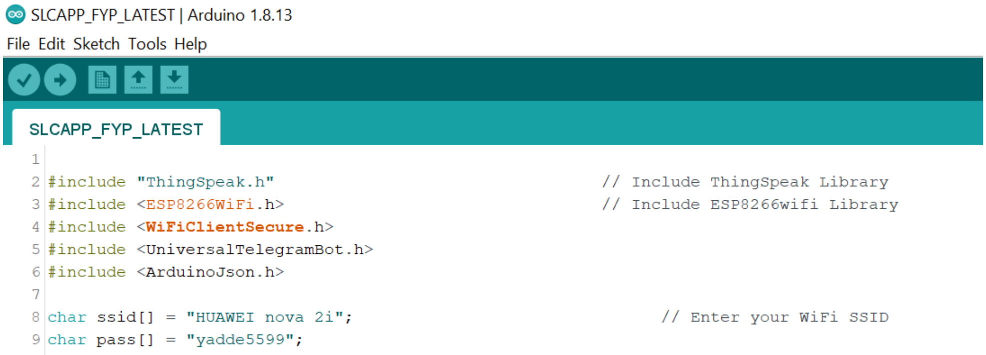


FYPTESTING | Arduino 1.8.13
File Edit Sketch Tools Help
FYPTESTING
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Get Board Info
Programmer
Burn Bootloader
Serial ports

Board: "NodeMCU 1.0 (ESP-12E Module)"
Builtin Led: "2"
Upload Speed: "115200"
CPU Frequency: "80 MHz"
Flash Size: "4MB (FS:2MB OTA:~1019KB)"
Debug port: "Disabled"
Debug Level: "None"
lwIP Variant: "v2 Lower Memory"
VTables: "Flash"
Exceptions: "Disabled (new can abort)"
Erase Flash: "Only Sketch"
SSL Support: "All SSL ciphers (most compatible)"
Port: "COM4"
✓ COM4

Figure 5.25: Microcontroller board selection

2. Next, write the code for the Node MCU ESP8266 microcontroller board in order to configure and define the library of Wi-Fi ESP8266 Module, ThingSpeak Module, Telegram bot and other few modules in the header.

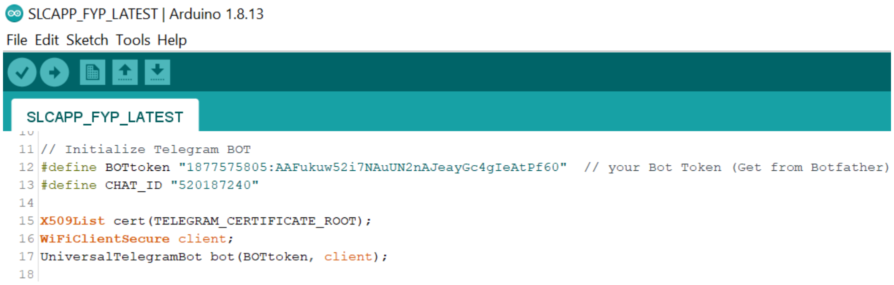


SLCAPP_FYP_LATEST | Arduino 1.8.13
File Edit Sketch Tools Help
SLCAPP_FYP_LATEST

```

#include "ThingSpeak.h"                                        // Include ThingSpeak Library
#include <ESP8266WiFi.h>                                        // Include ESP8266wifi Library
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>

char ssid[] = "HUAWEI nova 2i";                                       // Enter your WiFi SSID
char pass[] = "yadde5599";
```

Figure 5.26: Include the outside library

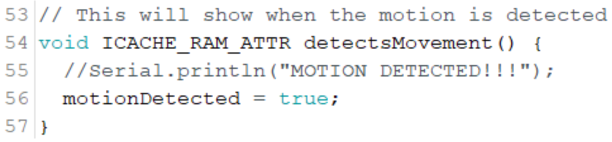
3. Next, in lines, 12 to 13 is the declaration of the Telegram Bot initialization of the telegram user chat id obtained from the IDbot and the BotToken generated earlier. Then, in line 15 to 16 is used to provide certificate root to the api.telegram.org and establish the bot.


SLCAPP_FYP_LATEST | Arduino 1.8.13
File Edit Sketch Tools Help
SLCAPP_FYP_LATEST

```
11 // Initialize Telegram BOT
12 #define BOTtoken "1877575805:AAFukuw52i7NAuUN2nAJeayGc4gIeAtPf60"  // your Bot Token (Get from Botfather)
13 #define CHAT_ID "520187240"
14
15 X509List cert(TELEGRAM_CERTIFICATE_ROOT);
16 WiFiClientSecure client;
17 UniversalTelegramBot bot(BOTtoken, client);
18
```

Figure 5.27: Bot establishment code

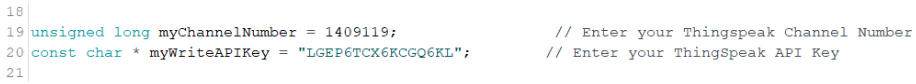
4. Then from lines 53 to 56, the code to detect movement where the detectMovement() method is being applied. This method is responsible for a callback, which is supposed to be executed when detecting any motion. The void ICACHE_RAM_ATTR is acting as a linker attribute and also to avoid any random crashes.



```
53 // This will show when the motion is detected
54 void ICACHE_RAM_ATTR detectsMovement() {
55   //Serial.println("MOTION DETECTED!!!");
56   motionDetected = true;
57 }
```

Figure 5.28: Bot movement detection code

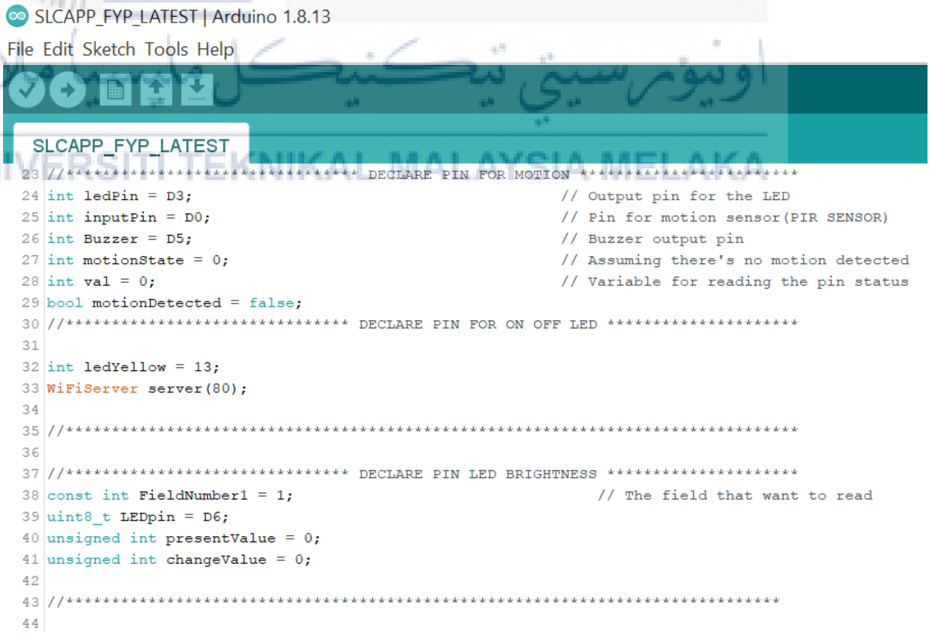
5. For the ThingSpeak connection, the channel number and the ThingSpeak api key need to be declared to obtain the real-time data from the hardware and keep it in the cloud.



```
unsigned long myChannelNumber = 1409119;              // Enter your Thingspeak Channel Number
const char * myWriteAPIKey = "LGEP6TCX6KCGQ6KL";        // Enter your ThingSpeak API Key

```

Figure 5.29: ThingSpeak declaration code

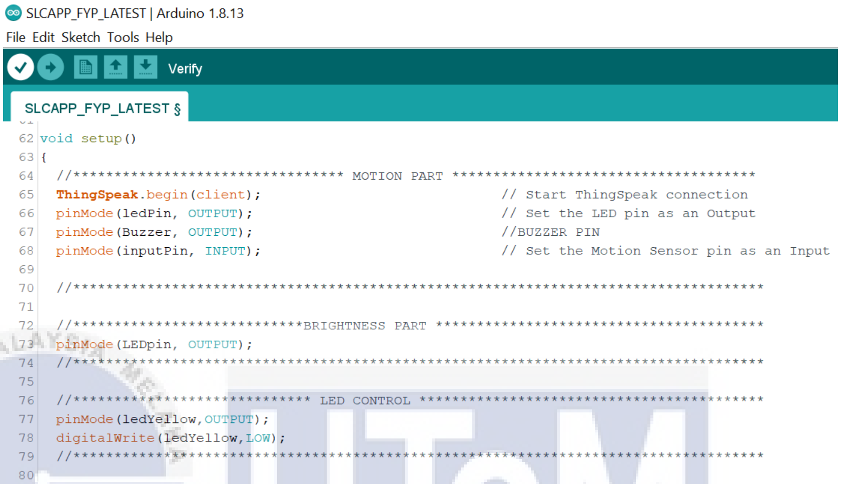
6. Then, start to declare each pin used on the Node MCU ESP8266 microcontroller board. This is to ensure and ease the process of knowing and understand which of the specified pin is belongs to either input or output. The pin used is distributed into 4 parts: the motion part, LED light control, and brightness part.


SLCAPP_FYP_LATEST | Arduino 1.8.13
File Edit Sketch Tools Help
SLCAPP_FYP_LATEST

```
//******************************** DECLARE PIN FOR MOTION **********************
int ledPin = D3;                                           // Output pin for the LED
int inputPin = D0;                                         // Pin for motion sensor(PIR SENSOR)
int Buzzer = D5;                                           // Buzzer output pin
int motionState = 0;                                       // Assuming there's no motion detected
int val = 0;                                               // Variable for reading the pin status
bool motionDetected = false;
//******************************* DECLARE PIN FOR ON OFF LED ********************

int ledYellow = 13;
WiFiServer server(80);

//********************************************************************************

//******************************* DECLARE PIN LED BRIGHTNESS ********************
const int FieldNumber1 = 1;                                    // The field that want to read
uint8_t LEDpin = D6;
unsigned int presentValue = 0;
unsigned int changeValue = 0;

//*******************************************************************************

```

Figure 5.30: Pin declaration on microcontroller

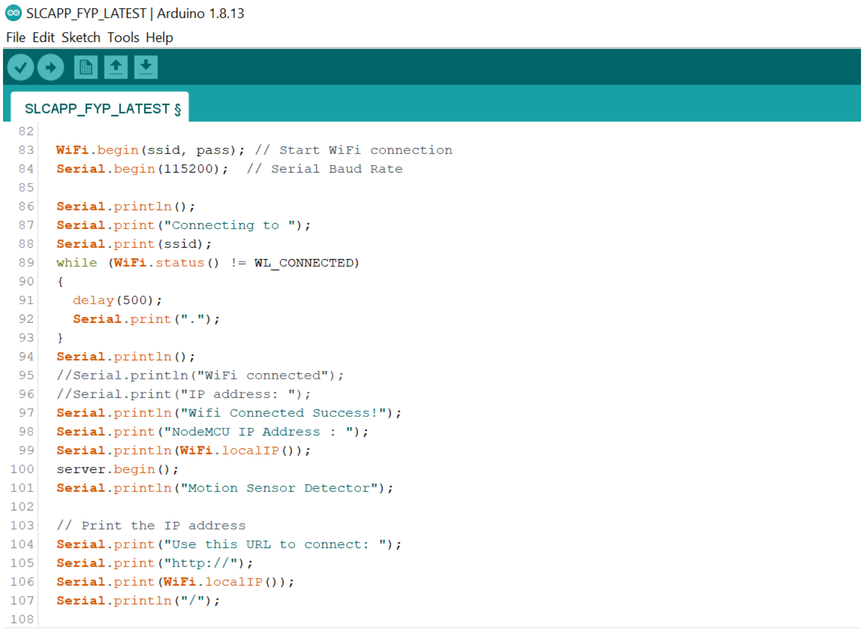
7. Next, will start with the void setup part where the setup () function is being used to set up the pin modes, initialize the variables declared, and allow access to the libraries imported previously. Hence, the void setup () section will only be run once as the microcontroller board is being powered.


SLCAPP_FYP_LATEST | Arduino 1.8.13
File Edit Sketch Tools Help
Verify
SLCAPP_FYP_LATEST §

```
void setup()
{
  //********************************* MOTION PART ************************************
  ThingSpeak.begin(client);                             // Start ThingSpeak connection
  pinMode(ledPin, OUTPUT);                              // Set the LED pin as an Output
  pinMode(Buzzer, OUTPUT);                              //BUZZER PIN
  pinMode(inputPin, INPUT);                             // Set the Motion Sensor pin as an Input

  //*****************************************************************************************

  //****************************BRIGHTNESS PART ****************************************
  pinMode(LEDpin, OUTPUT);
  //*****************************************************************************************

  //******************************* LED CONTROL ********************************************
  pinMode(ledYellow,OUTPUT);
  digitalWrite(ledYellow,LOW);
  //*****************************************************************************************
```

Figure 5.31: Void Setup declaration

8. Then, to connect to the Wi-Fi access point is as simple as entering the char ssid[] and char pass[] of the used network. This is to enable communication between the microcontroller board and also the devices used. Previously, the code has been declared in line 8 to 9 statement. Then use the Wi-Fi begins to function in order to initialize the Wi-Fi network and show the current status of the network.


SLCAPP_FYP_LATEST | Arduino 1.8.13
File Edit Sketch Tools Help
SLCAPP_FYP_LATEST §

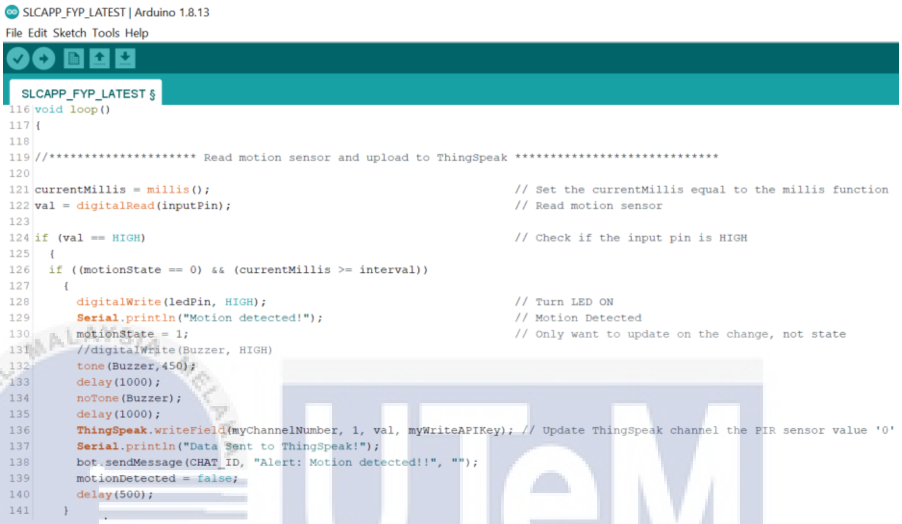
```

WiFi.begin(ssid, pass); // Start WiFi connection
Serial.begin(115200);  // Serial Baud Rate

Serial.println();
Serial.print("Connecting to ");
Serial.print(ssid);
while (WiFi.status() != WL_CONNECTED)
{
  delay(500);
  Serial.print(".");
}
Serial.println();
//Serial.println("WiFi connected");
//Serial.print("IP address: ");
Serial.println("Wifi Connected Success!");
Serial.print("NodeMCU IP Address : ");
Serial.println(WiFi.localIP());
server.begin();
Serial.println("Motion Sensor Detector");

// Print the IP address
Serial.print("Use this URL to connect: ");
Serial.print("http://");
Serial.print(WiFi.localIP());
Serial.println("/");

```

Figure 5.32: Wi-Fi connection coding

9. Next, in the void loop () coding part, the main code for the motion sensor and other functionality will be programmed. This is to ensure that it will react and respond repetitively. So, below shown is the motion sensor coding in the void loop () section, which starts from line 121 until line 153.

From line 121 to line 122, the motion sensor variable is being declared. It used the value 0 and 1 as the main value to an indication the motion status. So, the digitalWrite() method here plays the role of displaying output either it is HIGH or LOW. So, in line 124, it shows if the val == HIGH, it means that there is motion detected. Hence, the digitalWrite == HIGH and it will trigger the Piezo buzzer and also the LED pin status to HIGH.

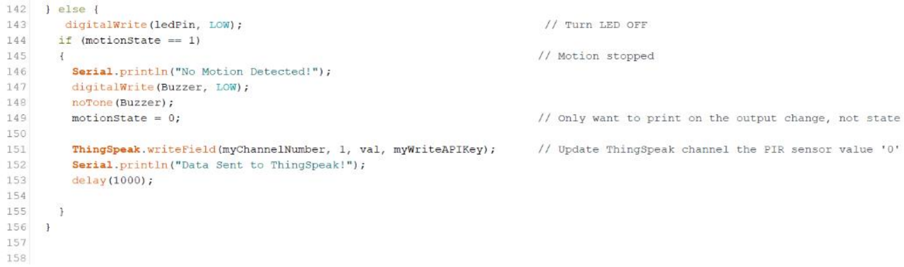
At the same time, the real-time data of motion will be retrieved and sent to the ThingSpeak channel simultaneously. It will be automatically updated every time motion is detected, as shown in lines 136 to 137. Meanwhile, the notification alert in lines 138 to 139 shows the code line on how the alert message is being sent to the user via the Telegram bot that has developed.


SLCAPP_FYP_LATEST | Arduino 1.8.13
File Edit Sketch Tools Help
SLCAPP_FYP_LATEST §

```
116 void loop()
117 {
118
119 //******************** Read motion sensor and upload to ThingSpeak ****************************
120
121 currentMillis = millis();                                             // Set the currentMillis equal to the millis function
122 val = digitalRead(inputPin);                                          // Read motion sensor
123
124 if (val == HIGH)                                                      // Check if the input pin is HIGH
125   {
126   if ((motionState == 0) && (currentMillis >= interval))
127     {
128       digitalWrite(ledPin, HIGH);                                     // Turn LED ON
129       Serial.println("Motion detected!");                             // Motion Detected
130       motionState = 1;                                                // Only want to update on the change, not state
131       //digitalWrite(Buzzer, HIGH)
132       tone(Buzzer,450);
133       delay(1000);
134       noTone(Buzzer);
135       delay(1000);
136       ThingSpeak.writeField(myChannelNumber, 1, val, myWriteAPIKey); // Update ThingSpeak channel the PIR sensor value '0'
137       Serial.println("Data Sent to ThingSpeak!");
138       bot.sendMessage(CHAT_ID, "Alert: Motion detected!!", "");
139       motionDetected = false;
140       delay(500);
141     }
```

Figure 5.33: Motion sensor coding in void loop

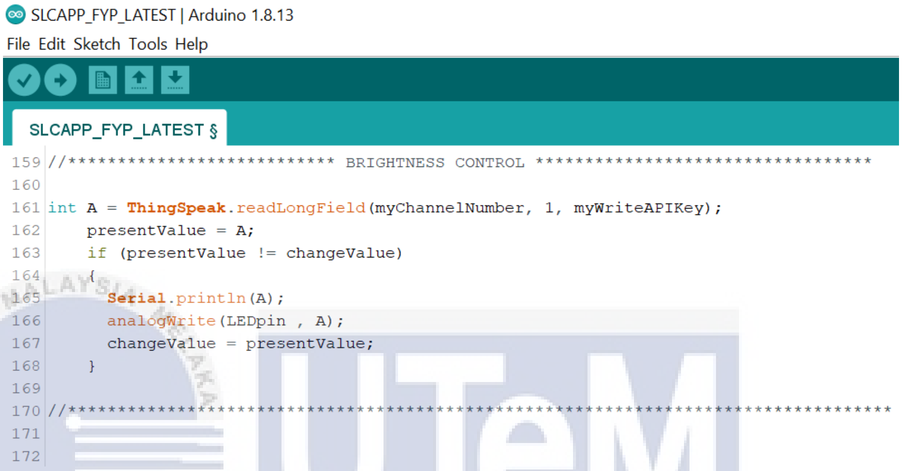
Otherwise, the digitalWrite == LOW means there is no motion detected by the PIR motion sensor, so the buzzer and the LED light will not be triggered. Hence, the real-time data will be sent to ThingSpeak as 0 value which indicates no motion identified.



```
} else {
   digitalWrite(ledPin, LOW);                                          // Turn LED OFF
  if (motionState == 1)
  {                                                                   // Motion stopped
    Serial.println("No Motion Detected!");
    digitalWrite(Buzzer, LOW);
    noTone(Buzzer);
    motionState = 0;                                                  // Only want to print on the output change, not state

    ThingSpeak.writeField(myChannelNumber, 1, val, myWriteAPIKey);    // Update ThingSpeak channel the PIR sensor value '0'
    Serial.println("Data Sent to ThingSpeak!");
    delay(1000);

  }
}
```

Figure 5.34: Motion sensor code if no motion detected

10. Next, the brightness level control functionality is being implemented with the integration of the ThingSpeak platform. As in this coding section, the analogWrite() method is being used, unlike the other functionality that used the DigitalWrite() method approach where produce the outcome of HIGH and LOW for some reason.

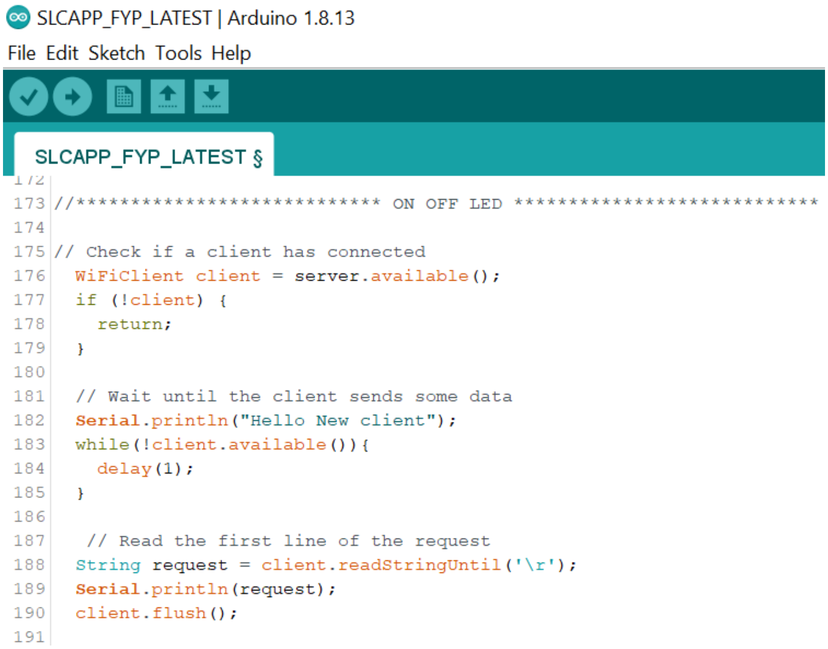

SLCAPP_FYP_LATEST | Arduino 1.8.13
File Edit Sketch Tools Help
SLCAPP_FYP_LATEST §

```
159 //*************************** BRIGHTNESS CONTROL **********************************
160
161 int A = ThingSpeak.readLongField(myChannelNumber, 1, myWriteAPIKey);
162     presentValue = A;
163     if (presentValue != changeValue)
164     {
165       Serial.println(A);
166       analogWrite(LEDpin , A);
167       changeValue = presentValue;
168     }
169
170 //*************************************************************************************
171
172
```

Figure 5.35: Brightness level control code

This is because the analog is the one that can produce value output in a PWM wave form. So, the brightness level can be controlled and changed according to the value itself, as shown in lines 162 to 167.

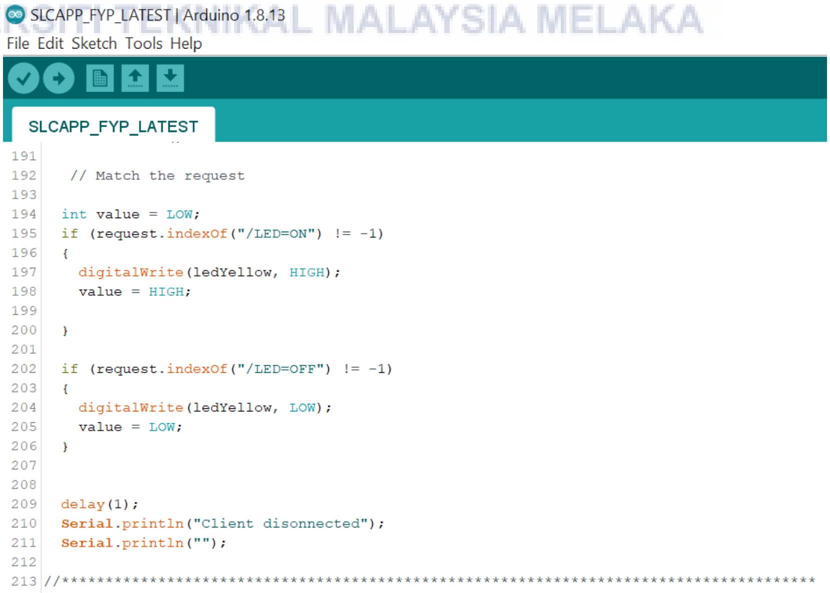
11. Then, for the on/off LED light control, the coding line from 175 to 190 is the code that will be displayed on the serial monitor port “COM4” to indicate and receive the client connection status that is available. This method will help the user control the mobile app's LED light via integrating the mobile device and hardware components.


SLCAPP_FYP_LATEST | Arduino 1.8.13
File Edit Sketch Tools Help
SLCAPP_FYP_LATEST §

```
//****************************** ON OFF LED ******************************

// Check if a client has connected
  WiFiClient client = server.available();
  if (!client) {
    return;
  }

  // Wait until the client sends some data
  Serial.println("Hello New client");
  while(!client.available()){
    delay(1);
  }

   // Read the first line of the request
  String request = client.readStringUntil('\r');
  Serial.println(request);
  client.flush();

```

Figure 5.36: Establish client connection

12. Then, if it matches the request, there is a condition applied in which the variable value is HIGH equal to 1 digital value. It indicates the yellow LED light is ON. Meanwhile, suppose the request index value variable is LOW equal to 0 digital value. In that case, it indicates the yellow LED light is OFF.


SLCAPP_FYP_LATEST | Arduino 1.8.13
File Edit Sketch Tools Help
SLCAPP_FYP_LATEST

```

   // Match the request

  int value = LOW;
  if (request.indexOf("/LED=ON") != -1)
  {
    digitalWrite(ledYellow, HIGH);
    value = HIGH;

  }

  if (request.indexOf("/LED=OFF") != -1)
  {
    digitalWrite(ledYellow, LOW);
    value = LOW;
  }


  delay(1);
  Serial.println("Client disonnected");
  Serial.println("");

//****************************************************************************************
```

Figure 5.37: On/off LED light control code

5.4 Implementation Status

This sub-topic will be discussing the project development implementation progress in detail. This involves the main aspect, such as the components used and the duration taken in developing the project prototype.

Table 5.1: Implementation Status

No	Sections	Description	Duration
1.	Software to be used installation set up	Completing the installation and also the setting of the software used, which is the Arduino IDE with a port setting, Fritzing software	3 days
2.	Sketch the circuit diagram of the project prototype	By using the Fritzing software, sketch the circuit diagram for the project	2 days
3.	Assembling Hardware components and prototype development	Assemble the hardware components first: the Node MCU ESP8266, PIR motion sensor, Piezo buzzer, LED light, resistor, jumper wire, USB cable (Type-c), and a breadboard and all components used.	5 days
4.	Coding setup on the microcontroller board	Coding implementation for the Node MCU ESP8266 board in terms of the Wi-Fi connection part and the integration of the hardware components pin related to the board.	7 days

5.	Implementation code for the PIR motion sensor	Apply the code for the motion sensor pin and also its functionality detecting the movement.	10 days
6.	Implementation code for Piezo Buzzer and LED light.	This code will function in order to create an alert sound from the Piezo buzzer with the respective tone sound buzzer and trigger red light as an indicator of intrusion.	7 days
7.	Implementation code for LED on/off control and brightness level.	Apply the code for on/off light control functionality and the brightness level control.	7 days
8.	ThingSpeak configuration setup.	Create ThingSpeak channel for retrieving real-time data from the motion sensor and LED brightness level.	5 days
9.	Application development using MIT App Inventor	Develop Android mobile application involving interface and functionality through the code block. Establish the connection from other software such as the Node MCU ESP8266, ThingSpeak, and others into the application.	20 days
10.	Implementation code for Telegram bot installation development in Arduino IDE	Develop and generate Telegram bot for the project involving the connection of Telegram bot in Arduino IDE and bot installation package, Arduino Json in the IDE.	15 days

5.5 Conclusion

This chapter will show the implementation process of the project in terms of both hardware and software aspects. It also shows how the process is from one functionality to another through the coding writing, software application development, and integration of each of the hardware components.

CHAPTER 6: TESTING

6.1 Introduction

In this chapter, the project testing will be performed and conducted. The final project development will be tested in this testing phase. In addition, in the testing phase, it will focus on utilizing the microcontroller that communicates with the other hardware components and the software development part, which is the application established to display results.

Hence, testing is one of the methods of determining a product's efficiency, whether it requires improvement or is a success. The test phase will include creating a test plan, a test strategy, and a test design. The project will consist of testing and usability features.

6.2 Test Plan

In this section explains the test plan that will be employed in the project. It will consist of a few actions that will outline the project's scope. The purpose of this test plan is to guarantee that all of the objectives are met in general. The major purpose of the test plan is to document difficulties, deal with faults that occur during system operation, and identify areas that need to be improved and enhance the system to make it works well.

6.2.1 Test Organization

The Smart Light Control with Notification Alert is a prototype that has been designed as a low-cost solution alternative for the small-scale farmer in order to ease their farm management with a much easier application system that suits the new modern era technology which nowadays lots of various industries use IoT technology as a platform that facilitates and improve human life in many aspects.

6.2.2 Test Environment

Wi-Fi connection is one of the requirements needed as the prototype needs to perform its functionality such as sending the real-time data to the ThingSpeak for graph monitoring view, sending notifications alerts to the user via Telegram and also controlling the light brightness level.

6.2.3 Test Schedule

The test schedule identifies time durations or periods taken for the project to be completed. In fact, several errors and problems occurred during this procedure, which will take more time to resolve and will need to be checked during the implementation phase. As a result, the testing step will continue until the system is error-free. The use of time must be planned carefully and efficiently.

6.3 Test Strategy

The test strategy serves as the guidance for the project’s software development cycle, describing the testing method. Hence, this stage will outline the suitable testing approach used in this project. Thus, the end-user is required to install the application on their Android smartphone to obtain notifications, access the features, and view the detail on the smartphone. The figure below shows the prototype test strategy used to test in this section.

6.3.1 Classes of Tests

The purpose of this test is to ensure that the project prototype would be able to interact and integrate with the microcontroller, the Node MCU ESP8266, PIR motion sensor, Piezo Buzzer LED, resistors, cable so that it can work well. Hence, all of these components are linked together to make it functional and capable of storing data in a real-time database.

6.4 Test Design

As for the test, design is the section where it discusses the most important part of the testing phase because it involves putting the components and modules to the test to validate the system. The project's functionality can be improved by conducting this testing, allowing it to meet its objectives. To offer the perfect output of the system, all components and modules must have a satisfactory result. The workflow for system testing is shown below: -

6.4.1 Test Description

The project's test description was utilized to determine which parts needed to be tested. This section will go over the components and modules required to get a precise and effective output. The test scenarios from the project system are detailed in the table below: -

i) Testing the connectivity of the microcontroller of Node MCU ESP8266 with a laptop.

Table 6.1: Microcontroller ESP8266 connectivity

Test	Test the connectivity of Node MCU ESP8266 with a laptop.
Test Purpose	Identify any errors on the ESP8266 microcontroller board.
Test Environment	Test all the hardware components connection with the microcontroller board.
Precondition	• The Node MCU ESP8266 and laptop has been connected via USB adapter. • Arduino IDE software has been installed in the laptop earlier.
Test Setup	1) Launch the Arduino IDE Software. 2) Connect the ESP8266 microcontroller board to the laptop. 3) Then, upload the source code into the microcontroller board.
Expected Result	• The Arduino IDE software can detect the microcontroller board • Suppose when compiling and uploading the code, it should not show any error message on the serial monitor port com.
Error Message	None
Result	Pass

ii) Testing the connectivity of the Node MCU ESP8266 with PIR motion sensor.

Table 6.2: PIR Motion sensor function

Test	Test the connectivity of Node MCU ESP8266 with PIR motion sensor.
Test Purpose	To test the sensor ability in detecting motion of the small-scale farm area.
Test Environment	In order to run the situation test for sensor, the microcontroller board must be setup first.
Precondition	• The PIR motion sensor and the Node MCU ESP8266 has been connected before.
Test Setup	1) Connect the PIR motion sensor component to the microcontroller board using jumper wire. 2) Switch on the Node MCU ESP8266 microcontroller board 3) Upload and run the code program that function in order to detect the real-time movement or presence of human or any wild animal.
Expected Result	• The PIR motion sensor should be able to detect movement of living object such as human or animal. • Suppose the PIR Motion sensor will detect infrared radiation that emitted from the entity itself and send the real time data representative which are 1 and 0 to the ThingSpeak.
Error Message	None
Result	Pass

iii) Testing the Piezo Buzzer function

Table 6.3: Buzzer function testing

Test	Buzzer sound function test
Test Purpose	To test the buzzer of the prototype that function to produce alert sound.
Test Environment	In order to run the situation test for the buzzer, the microcontroller board must be setup first, as in the section 5.2.
Precondition	• The Piezo buzzer and the Node MCU ESP8266 have been connected before.
Test Setup	1) Connect the Piezo buzzer component to the microcontroller board using jumper wire. 2) Switch on the Node MCU ESP8266 microcontroller board 3) Upload and run the code program that function to produce an alert sound as an output when detecting any motion and trigger the red LED light as a precaution indicator.
Expected Result	• The buzzer will be able to produce a high tone buzzer sound as for warning alert when detecting motion and trigger the red-light LED.
Error Message	None
Result	Pass

iv) Testing the LED function

Table 6.4: LED function testing

Test	LED function test
Test Purpose	To test the LED light of the prototype for the Turn on/off and brightness control functionality.
Test Environment	In order to run the situation test for the led, the microcontroller board must be setup first, as in the section 5.2.
Precondition	• The LED and the Node MCU ESP8266 have been connected before.
Test Setup	1) Connect the LED component to the microcontroller board using the breadboard and resistor. 2) Switch on the Node MCU ESP8266 microcontroller board 3) Upload and run the code program to the microcontroller board. 4) Turn on and off the LED light via the Android smartphone application. 5) Control the amount of the brightness level of the LED.
Expected Result	• The LED light will function as after the source code has been uploaded. The user can manage to turn on or off the LED light and control the brightness level amount.
Error Message	None
Result	Pass

v) Testing the ThingSpeak function

Table 6.5: ThingSpeak function

Test	ThingSpeak function test
Test Purpose	To test the ThingSpeak functionality and connectivity to the Node MCU ESP8266 microcontroller board.
Test Environment	In order to run the situation test for ThingSpeak need to launch the microcontroller board setup first and the other component as well.
Precondition	• The ThingSpeak has been setup and configured before.
Test Setup	1) Create the ThingSpeak channel on the website for the Motion detection graph. 2) Switch on the Node MCU ESP8266 microcontroller board 3) Upload and run the code program to the microcontroller board. 4) After the motion sensor detect the motion, it will send the real-time data to ThingSpeak. 5) ThingSpeak will store the data representative and display on the graph view.
Expected Result	• The ThingSpeak will display the current real-time data view of the motion sensing detection details whether 1 or 0 detection status including it timestamp and date.
Error Message	None
Result	Pass

vi) Testing the notification function on smartphone

Table 6.6: Notification function testing

Test	Notification on Telegram bot function test
Test Purpose	To test the user will receive the notification alert message on the Telegram of the warning alert if motion detected.
Test Environment	In order to run the situation test for notification the Telegram bot and the microcontroller board must be first setup.
Precondition	• The Telegram bot has been setup and configured before.
Test Setup	1) Create the Telegram chatbot application and obtain the token as to integrate with the microcontroller board. 2) Upload and run the code program of the Telegram chatbot setting on the Arduino IDE to the microcontroller board 3) If any motion is detected, a popup notification alert message will be automatically sent to the user indicate the alert warning.
Expected Result	• The alert notification will be sent to the user via the Telegram chatbot that has been setup automatically after the motion sensor been triggered.
Error Message	None
Result	Pass

i) Testing the weather forecast information view

Table 6.7: Weather forecast testing

Test	Weather forecast information testing
Test Purpose	To test the user will able to view the current weather temperature climate according to their current location in real-time view.
Test Environment	In order to run the situation, test the SLC app that has been created using MIT App Inventor must be open.
Precondition	• The mobile application that has been fully developed before.
Test Setup	1) Create the application interface for weather forecast menu 2) Integrate the weather API website to the application develop
Expected Result	• User will be able to view the current weather temperature daily anytime, anywhere according to their current location.
Error Message	None
Result	Pass

6.4.2 Test Data

The purpose of the data testing is it used to verify the data on the project. As a result, the test data is being used to determines whether it will achieve the project deliverables or not.

1. Node MCU ESP8266 connectivity test

The figure below shows the connectivity test of the Node MCU ESP8266 microcontroller board that is connected to the laptop via a USB cable.

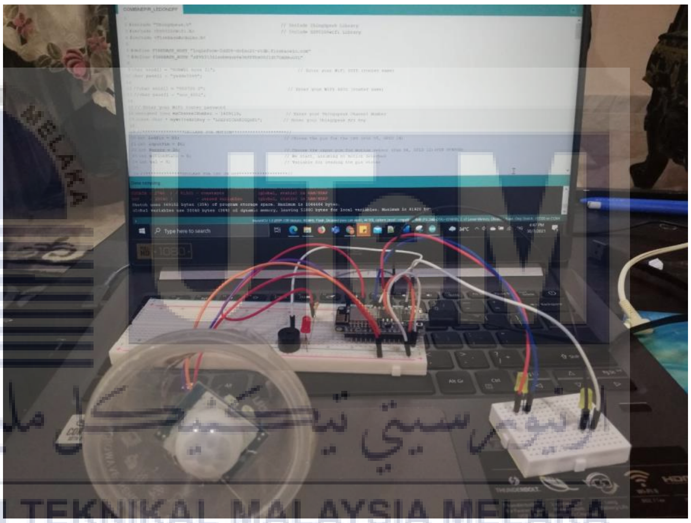
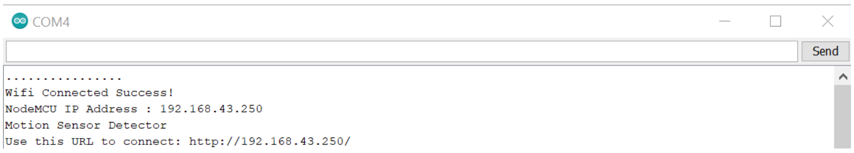


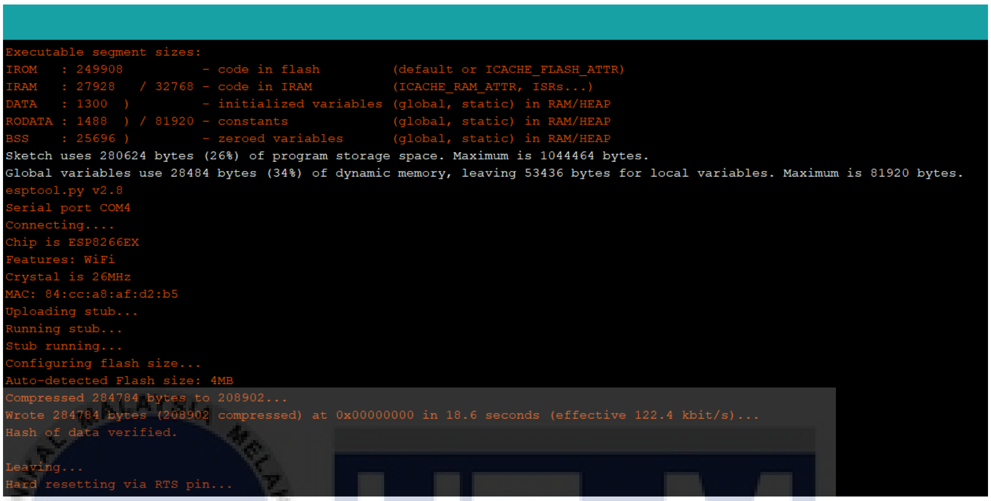
Figure 6.1: Microcontroller connectivity testing

Therefore, to identify whether the connectivity status is successful or not, thus it can be a check on the Arduino IDE via serial monitor port COM4 and run the code. The figure below shows a connection between the Node MCU ESP8266 and the USB port by listing the port com for code upload purposes.



```
................
Wifi Connected Success!
NodeMCU IP Address : 192.168.43.250
Motion Sensor Detector
Use this URL to connect: http://192.168.43.250/
```

Figure 6.2: Serial Monitor port COM4



```
Executable segment sizes:
IROM   : 249908          - code in flash         (default or ICACHE_FLASH_ATTR)
IRAM   : 27928   / 32768 - code in IRAM          (ICACHE_RAM_ATTR, ISRs...)
DATA   : 1300  )         - initialized variables (global, static) in RAM/HEAP
RODATA : 1488  ) / 81920 - constants             (global, static) in RAM/HEAP
BSS    : 25696 )         - zeroed variables      (global, static) in RAM/HEAP
Sketch uses 280624 bytes (26%) of program storage space. Maximum is 1044464 bytes.
Global variables use 28484 bytes (34%) of dynamic memory, leaving 53436 bytes for local variables. Maximum is 81920 bytes.
esptool.py v2.8
Serial port COM4
Connecting....
Chip is ESP8266EX
Features: WiFi
Crystal is 26MHz
MAC: 84:cc:a8:af:d2:b5
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Auto-detected Flash size: 4MB
Compressed 284784 bytes to 208902...
Wrote 284784 bytes (208902 compressed) at 0x00000000 in 18.6 seconds (effective 122.4 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

Figure 6.3: Connectivity console status

2. Component Connectivity Test

The figure below shows the connection setup of the hardware components: the PIR motion sensor, Buzzer, LED lights, and some other components with the ESP8266 board connected via the jumper wires and the breadboard.

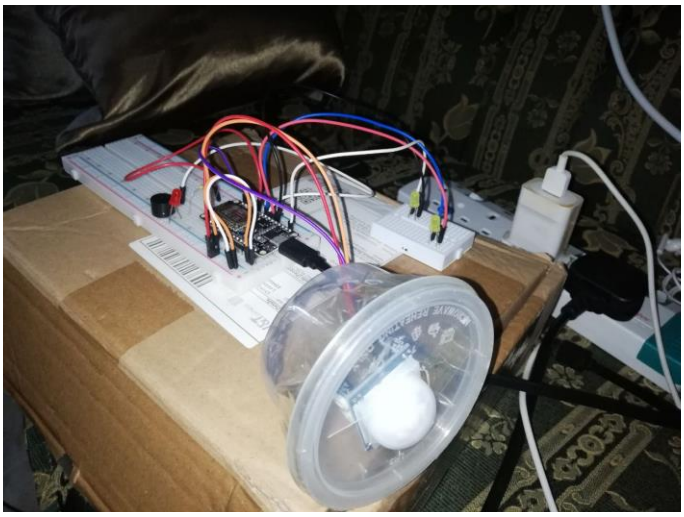


Figure 6.4: Component connectivity setup

The figure 6.5 below shows the components connected is functioning perform a respective task which will facilitate the user to manage to use easily with the good connection.

Figure 6.5: Motion detection component response to the functionality

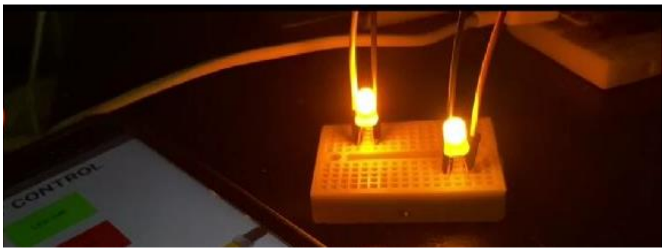


Figure 6.6: LED light response to its functionality

3. Application connectivity test

For this project application development, the MIT App Inventor platform has been used in order to establish the Smart Light Control application. Hence, the application has been named the SLC app and used on Android smartphones.

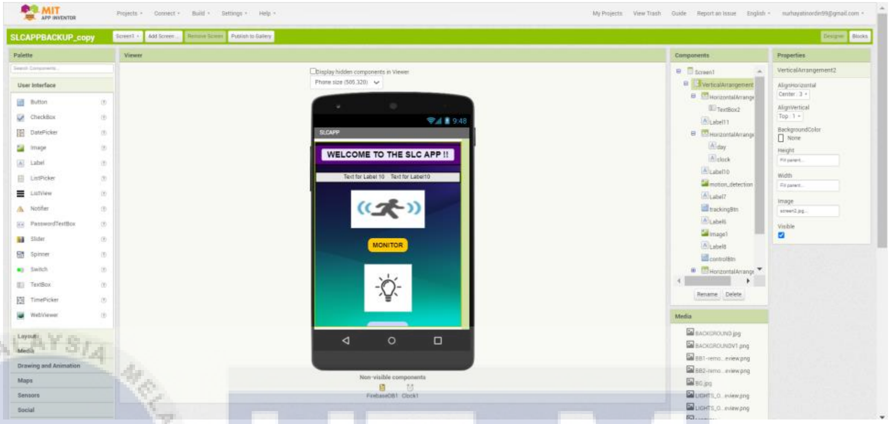


Figure 6.7: MIT Apps

The figure shown below is the starting screen of the application where this screen will appear first and then will be redirected to the menu list page view.

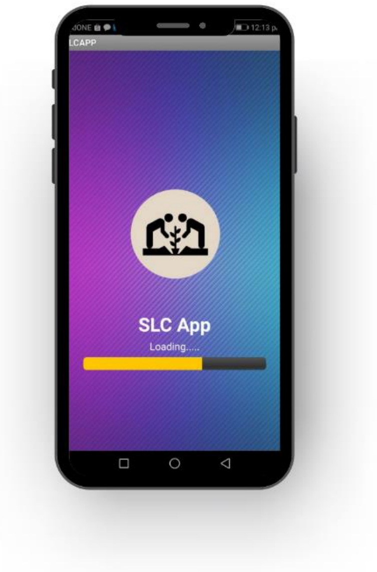


Figure 6.8: Starting screen of the application

The first interface of this project is the application's view list menu, which consists of 3 menus as shown in the diagram. Each and every one of these menus will be redirected to its responsive page according to the menu tsk itself, such as monitoring the motion detection, light control page, and weather forecast information view page.

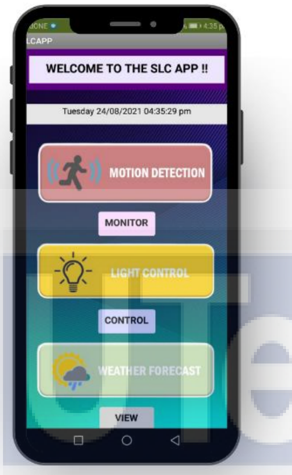


Figure 6.9: First Interface

As for this project's second interface, the ThingSpeak is integrated with the application through the API created. The figure below shows the real-time data obtain from the motion sensor updated in the real-time graph view with the details of the timestamp and condition status data representative.

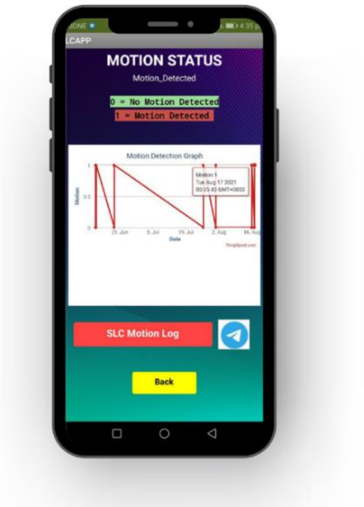


Figure 6.10: Second Interface of the application

The third interface of this project is the light controlling page, where users can either perform light on/off or brightness control activity via their smartphone. The figure below shows the 2-button available for the light on/off and the slider widget to control the light brightness.

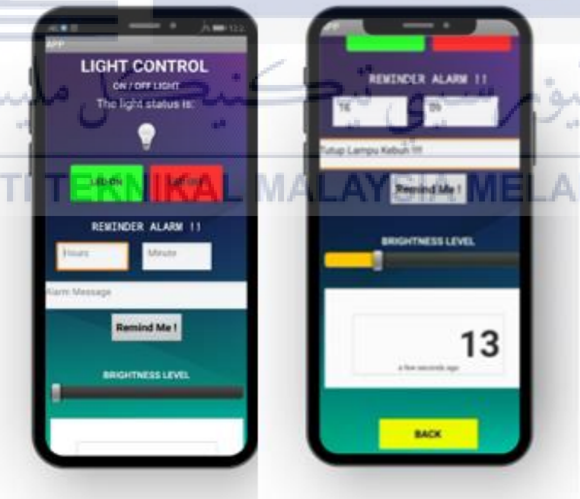
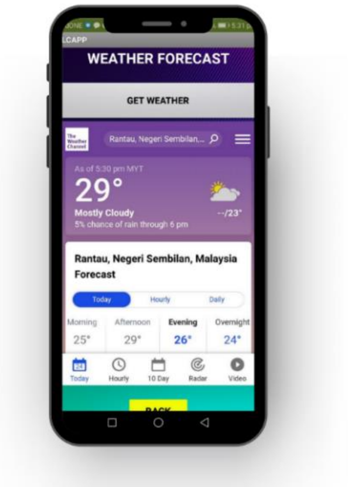


Figure 6.11: Third Interface of the application

The fourth interface is the forecast weather information which is one of the additional features that can ease the farmer in order to estimate the weather of the day. The forecast weather may help the farmer to plan their daily farming activity by just press the get weather button to view the current weather condition of the location.



Rantau, Negeri Sembilan,...

As of 5:30 pm MYT

29°

Mostly Cloudy

5% chance of rain through 6 pm

--/23°

Rantau, Negeri Sembilan, Malaysia Forecast

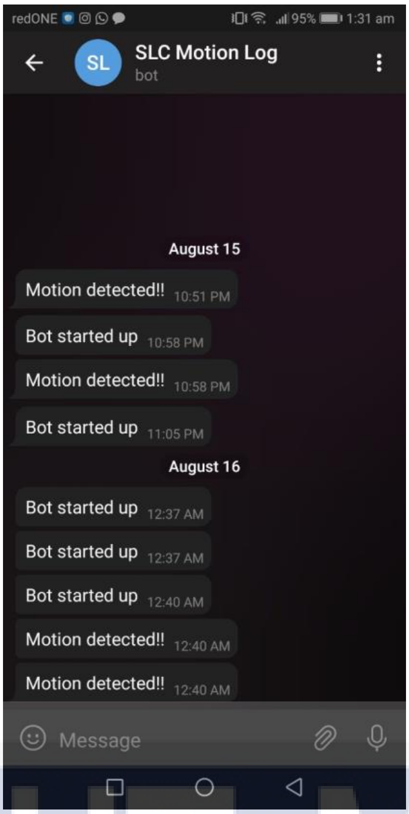
Morning	Afternoon	Evening	Overnight
25°	29°	26°	24°

Figure 6.12: Fourth Interface of the application

The figure above shows the current weather information of the location to ease the small-scale farmer in making plans and decisions to carry out their agricultural activities well and smoothly every day.

4. Notification Alert test

The figure shown below displays the notification alert message sent to the user as the motion detected via the Telegram bot. AS after the bot is automatically startup, the message will also automatically send to the user stated that there is motion detected at their farm area.


SLC Motion Log
bot

August 15

Motion detected!! 10:51 PM

Bot started up 10:58 PM

Motion detected!! 10:58 PM

Bot started up 11:05 PM

August 16

Bot started up 12:37 AM

Bot started up 12:37 AM

Bot started up 12:40 AM

Motion detected!! 12:40 AM

Motion detected!! 12:40 AM

Figure 6.13: Notification Alert on Telegram

6.5 Test Result and Analysis

Following are the completion of the previously described test plan, test strategy, and test design. Once the project is fully integrated, test cases and the expected outcome of the entire project will be recorded. The testing results on each of the components utilized in this project are shown in the sections below.

6.5.1 Test Result on Hardware

i. PIR motion sensor function

Table 6.8: Motion sensor result analysis

Test Case Identification	1	2
Test Identification	Sensor detecting motion	Sensor doesn't detect any motion
Result Expectation	Thus, if any motion of living object such as human is detected it will send the real-time data to the ThingSpeak as a motion indicator via representative data which is 1.	If there is no motion detected the sensor will send the real-time data representative which is 0 to the Thingspeak as no motion detected.
Success/Fail	Success	Success

The figure shown above is about the testing summarization of the PIR Motion sensor function that has been conducted. Hence, this result analysis is needed to ensure that all the components respond when conducting the test. The motion sensor will help the farmer in order to detect any suspicious movement from the outside of the small-scale farm region.

ii. Piezo Buzzer function

Table 6.9: Piezo Buzzer function result analysis

Test Case Identification	1	2
Test Identification	Sensor detecting motion	Sensor doesn’t detect any motion
Result Expectation	The Piezo buzzer will be triggered and produce a high tone buzzer sound and the red-light LED will be turn on too as an alert indicator.	The Piezo buzzer will not produce any sound and the red-light LED will be remain off.
Success/Fail	Success	Success

The figure shown above is about the testing summarization of the PIR Motion sensor function that has been conducted. Hence, this result analysis is needed to ensure that all the components respond when conducting the test. The alert sound may help notify the farmer of impending threats in the small-scale farm area if any intrusion happens.

iii. LED light on/off function

Table 6.10: LED light on/off function result analysis

Test Case Identification	1	2
Test Identification	The light on button was pressed	The light off button was pressed
Result Expectation	The LED light will be turn on when the on button is being pressed and control via the mobile application	The LED light will be turn off when the off button is being pressed on mobile application.

	on their Android Smartphone.	
Success/Fail	Success	Success

The figure above shows the testing summarization of the on/off led light function that has been conducted. Hence, this result analysis is needed to ensure that all the components respond when conducting the test. The on/off control function may help the farmer control the lights when they are out of the region as they can simply control via their Android mobile application.

iv. LED brightness level control

Table 6.11: Brightness control function level analysis

Test Case Identification	1	2
Test Identification	The light brightness level control adjustment	The light brightness level control doesn't have any adjustment
Result Expectation	The LED light brightness will be appeared according to the amount of brightness level control on the mobile application.	The LED light will be off or became dimmer if the amount of brightness level control decreases.
Success/Fail	Success	Success

The figure shown above is about the testing summarization of the light brightness level control function that has been conducted. Hence, this result analysis is needed to ensure that all the components respond when conducting the test. This function may help the farmer control the light's brightness according to how bright the light they want depends on the amount of brightness level on the widget.

6.5.2 Test Result on Application

i. MIT Apps

Table 6.12: Application test function

Test Case Identification	1	2
Test Identification	User receive real-time data of the motion detection on ThingSpeak	User receive real-time data is no motion detection on ThingSpeak
Result Expectation	The user will able to view the real-time data as 1 of motion detection as the ThingSpeak will update the status on the motion detection graph including its timestamp and date.	The user will able to view the real-time data as 0 of motion detection as the ThingSpeak will update the status on the motion detection graph including its timestamp and date.
Success/Fail	Success	Success

The figure shown above is about the testing summarization of real-time data graph view by using ThingSpeak. As for this functionality, the data will be kept updated every time it obtains any motion detection data in real-time based. Hence, users can monitor via the motion detection graph on their Android smartphone.

ii. Notification alert on Telegram

Table 6.13: Notification function result analysis

Test Case Identification	1	2
Test Identification	User receive notifications alert	User not receive any notifications alert
Result Expectation	User will receive a pop-up notification on the Telegram stating alert message if there is any suspicious movement detected in the farm region.	User will not receive any notifications on the Telegram stating alert message if there is any suspicious movement detected in the farm region.
Success/Fail	Success	Success

The figure shown above is about the testing summarization of notification alerts to the user via the Telegram platform. As for this functionality, the notification alert will be automatically sent to the user's smartphone every time it obtains any motion detection data in real-time based.

6.6 Conclusion

In a conclusion, all of the functionality tests and outcome evaluations were completed in this stage. The feature test is used to ensure that the prototype can work properly. The findings were evaluated and concluded based on different settings. The conclusion of this project is discussed in the following section.

CHAPTER 7: CONCLUSION

7.1 Introduction

This chapter will summarize the overall aspect of the project from the beginning until the end of the project plan. Hence, it will also highlight the project's limitations as well as potential work for future enhancements. Thus, it will increase the scheme's efficiency and efficacy, making it more effective and thorough.

7.2 Project Summarization

Smart Light Control with Notification Alert for a small-scale farm is a project of low-cost solution with the IoT technology approach for the farmer. The purpose of this project is to facilitate proper and easy management in enhancing the security level of their farm area via a simple API development application. As for this project, there are three (3) objectives stated in Chapter 1 that need to be achieved. There are, first to develop a low-cost IoT solution prototype that can automatically detect the motion and control the light, Second, to notify the user through the notification alert as a precaution and last, to test the functionality of the prototype of the project.

Furthermore, this project may reduce the burden of these small-scale farmers to control or manage their farm when they are out if the region with the aid of this simple project where they don’t have to be worried as they can just control and monitor via their Android smartphone application. As for the CCTV installation are way much expensive, involving its installation cost, needed high bandwidth usage to load the video footage for analysis, lead to excessive money spending on the hardware setup, especially for the surveillance camera, PC monitor and so on which is not convenient to be used by a single small-scale farmer.

7.3 Project Contribution

This section, which is the project contribution, will describe or detail what kind of aspects the project can contribute to the user. Thus, this Smart Light Control with Notification Alert for the small-scale farm can somehow benefit the users, especially in term of cost where it is very affordable and reasonable, which is not costly in terms of installation setup and includes its characteristic, user-friendly. In fact, Smart Light control with Notification Alert for Small-Scale farms can be a good low-cost solution to the human life as everything is just at one’s fingertips, where they can just control and monitor the farm condition via the smartphone at any time, anywhere and also access all the details by just using the smartphone. Hence, they can also obtain notifications alert if any suspicious motion is detected into the farm if they were outside the farm region with a warning alarm as an intrusion indicator.

7.4 Project Limitation

This section's purpose is to highlight any project limitations, such as the project constraints or restrictions. This section also examines several circumstances that may cause the project's objectives to be lowered and how to recognize the project's limitations. As for this project, the project limitation is that the PIR motion sensor might have to struggle if the weather temperature is > 35°C especially in the hotter climate that might impact the sensor. But as in Malaysia, the normal average climate weather is in the range of 25°C to 27°C it should be fine. Hence, the Internet connection is also one of the project limitations in this project as it is using the Node MCU ESP8266 where it needs a Wi-Fi connection in order to perform the code that will be uploaded into the microcontroller board and for a user to receive notification alert, retrieve real-time motion sensor data requires internet connection. So, if the Internet connection is out of range, there will be no data will be received.

7.5 Future Works

In the future, this project can be enhanced in terms of its application features or the development of any new implementation to make it more functional and detailed. Some new enhancements that can be made can improve the application GUI into a more elegant and systematic interface where can add more other functional widgets like the farm plant condition if any unknown damage has been done to the farm crop yields. Therefore, with GUI improvement, it hopes it can increase efficiency and improve the application's user-friendliness. Next, in terms of the buzzer, that can be improved by enhancing it into a high decibel alarm that can produce a more louder sound alert for the project.

7.6 Conclusion

At the end of this project, starting from the planning, analysis, design, implementation, and testing phase, the Smart Light Control with Notification Alert for a small-scale farm is successfully completed. Thus, all of the project objectives have been achieved. Hopefully, this project development might give some benefits and facilitate the user in lessening the burden in managing their small-scale farm area condition in their daily routine. In addition, this project is an alternative approach as it is a low-cost solution and user-friendly motion detector that can afford with a simple application.

REFERENCES

R. Leandros, W. J. Widjaja Saputra, D. Fitria Murad, D. Atmaja August 2020. SMS Notification On-Off Room Lights with Body Detection Using Microcontrollers.

S. Yadahalli, A. Parmar, Prof. A. Deshpande (2020). Smart Intrusion Detection System for Crop Protection by using Arduino.

S. Santhiya, Y. Dhamodharan, N. E. Kayi Priya, C. S. Santhosh, M. Surekha March (2018). A smart Farmland Using Raspberry Pi Crop Prevention and Animal Intrusion Detection System.

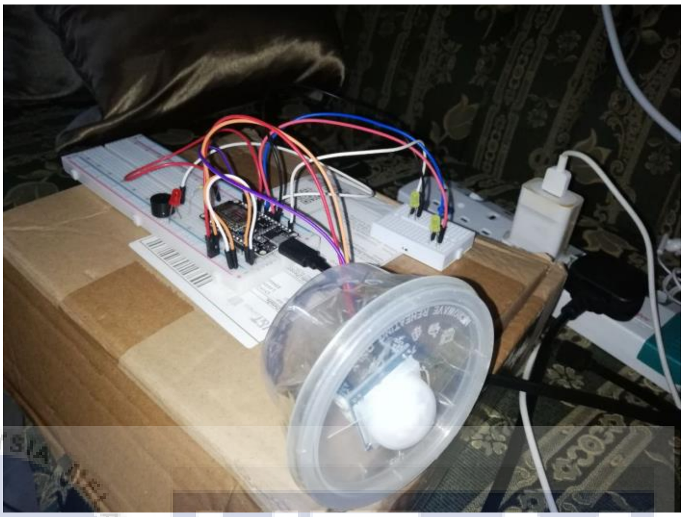
Nilesh A. Lakade1, Rajat Patel, Pinak Panchal, (2019, June). Intelligent Lighting System for Classrooms and Mall using IoT. International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 06 Issue

Syeda Puspita Mouri, Syed Nazmus Sakib, Zannatul Ferdous, Md. Abu Taher, (2016, January). Automatic Lighting and Security System Design using PIR Motion Sensor. IEEE

Deepak Kumar Rath, (2016, March-April). Arduino Based: Smart Light Control System. International Journal of Engineering Research and General Science Volume 4, Issue 2.

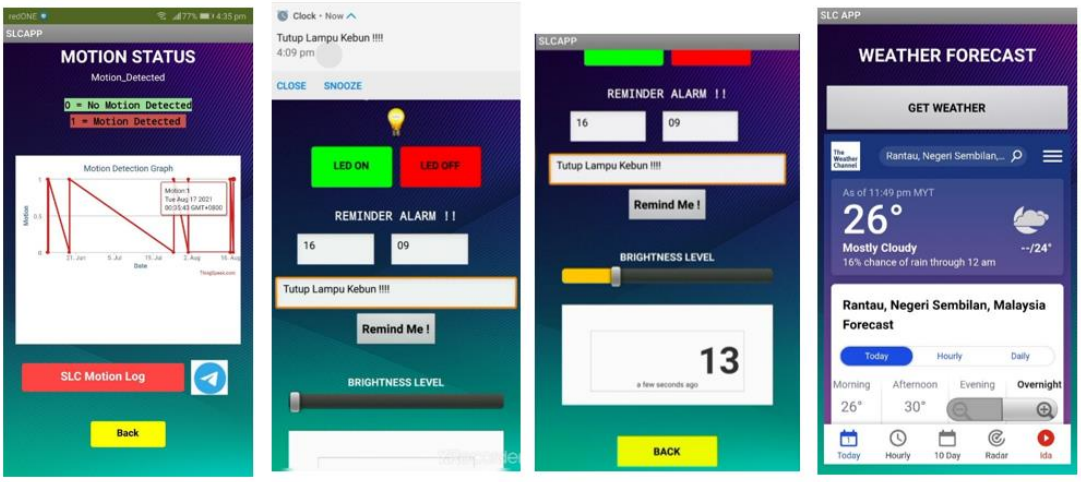
A.Z.M Tahmidul Kabir, Al Mamun Mizan, Nirmal Debnath, Akib Jawad Ta-sin, N. Zinnurayen, Md. Tanhir Haider (2020). IoT Based Low-Cost Smart Indoor Farming Management System Using an Assistant Robot and Mobile App.

APPENDICES

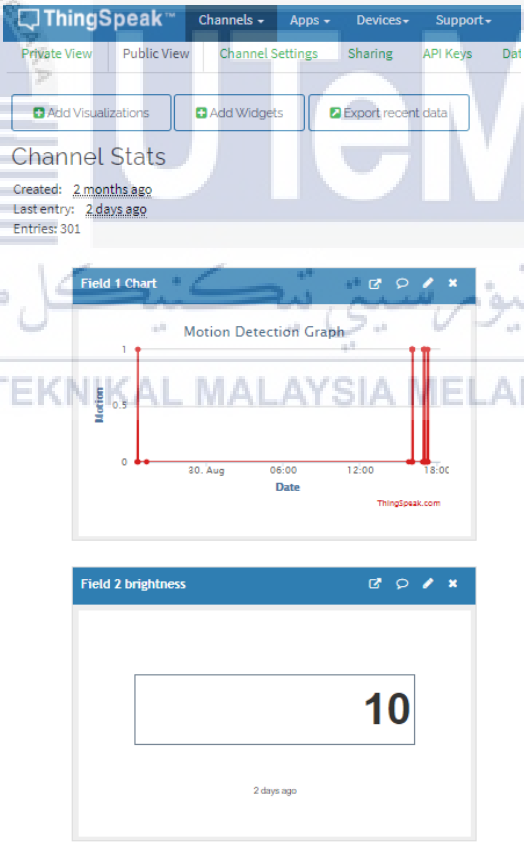


Appendix A: Prototype progress development

Appendix B: Prototype completed testing



Appendix C: Application testing display



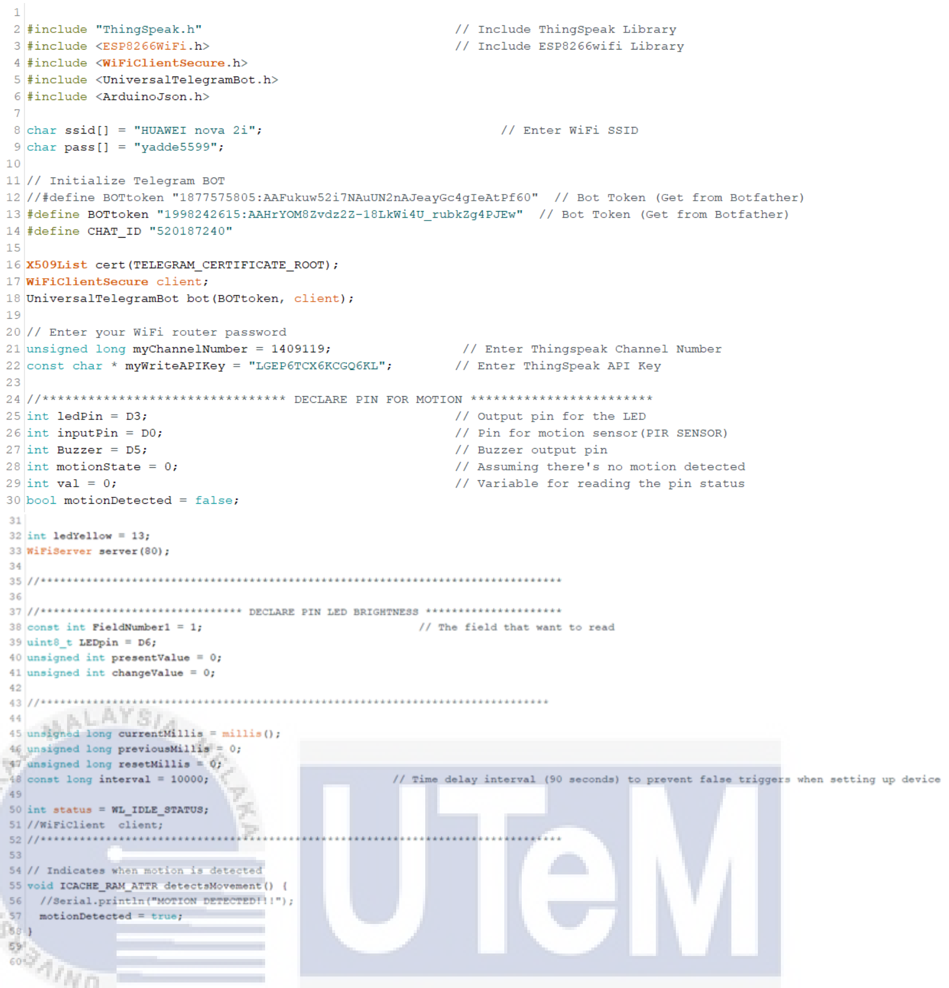
Channel Stats

Created: 2 months ago

Last entry: 2 days ago

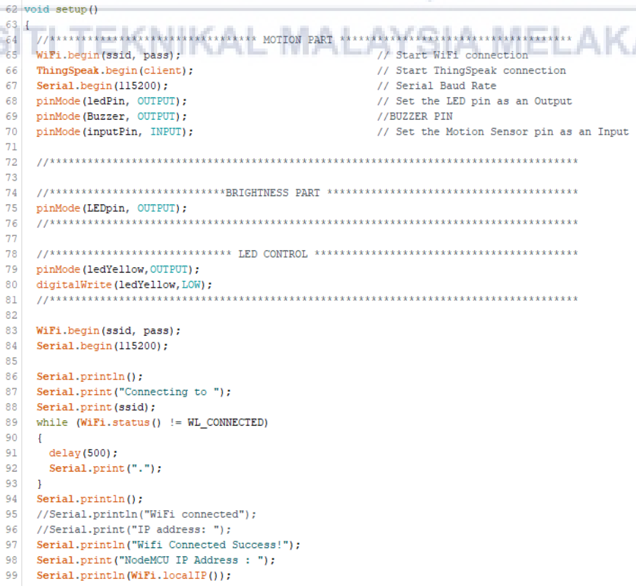
Entries: 301

Appendix D: real-time data sent to the ThingSpeak Cloud



```

#include "ThingSpeak.h"                                  // Include ThingSpeak Library
#include <ESP8266WiFi.h>                                 // Include ESP8266wifi Library
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>

char ssid[] = "HUAWEI nova 2i";                                // Enter WiFi SSID
char pass[] = "yadde5599";

// Initialize Telegram BOT
//#define BOTtoken "1877575805:AAFukuw52i7NAuUN2nAJeayGc4gIeAtPf60"  // Bot Token (Get from Botfather)
#define BOTtoken "1998242615:AAHrYOM8Zvdz2Z-18LkWi4U_rubkZg4PJEw"  // Bot Token (Get from Botfather)
#define CHAT_ID "520187240"

X509List cert(TELEGRAM_CERTIFICATE_ROOT);
WiFiClientSecure client;
UniversalTelegramBot bot(BOTtoken, client);

// Enter your WiFi router password
unsigned long myChannelNumber = 1409119;                  // Enter Thingspeak Channel Number
const char * myWriteAPIKey = "LGEP6TCX6KCGQ6KL";         // Enter ThingSpeak API Key

//********************************** DECLARE PIN FOR MOTION ************************
int ledPin = D3;                                         // Output pin for the LED
int inputPin = D0;                                       // Pin for motion sensor(PIR SENSOR)
int Buzzer = D5;                                         // Buzzer output pin
int motionState = 0;                                     // Assuming there's no motion detected
int val = 0;                                             // Variable for reading the pin status
bool motionDetected = false;

int ledYellow = 13;
WiFiServer server(80);

//*******************************************************************************

//******************************** DECLARE PIN LED BRIGHTNESS ********************
const int FieldNumber1 = 1;                              // The field that want to read
uint8_t LEDpin = D6;
unsigned int presentValue = 0;
unsigned int changeValue = 0;

//******************************************************************************

unsigned long currentMillis = millis();
unsigned long previousMillis = 0;
unsigned long resetMillis = 0;
const long interval = 10000;                             // Time delay interval (90 seconds) to prevent false triggers when setting up device

int status = WL_IDLE_STATUS;
//WiFiClient  client;
//*******************************************************************************

// Indicates when motion is detected
void ICACHE_RAM_ATTR detectsMovement() {
  //Serial.println("MOTION DETECTED!!!");
  motionDetected = true;
}


```

Appendix E: Arduino source code for pin declaration and package setup



```
void setup()
{
  //****************************** MOTION PART **********************************
  WiFi.begin(ssid, pass);                                // Start WiFi connection
  ThingSpeak.begin(client);                              // Start ThingSpeak connection
  Serial.begin(115200);                                  // Serial Baud Rate
  pinMode(ledPin, OUTPUT);                               // Set the LED pin as an Output
  pinMode(Buzzer, OUTPUT);                               //BUZZER PIN
  pinMode(inputPin, INPUT);                              // Set the Motion Sensor pin as an Input

  //*****************************************************************************

  //****************************BRIGHTNESS PART *********************************
  pinMode(LEDpin, OUTPUT);
  //*****************************************************************************

  //***************************** LED CONTROL ***********************************
  pinMode(ledYellow,OUTPUT);
  digitalWrite(ledYellow,LOW);
  //*****************************************************************************

  WiFi.begin(ssid, pass);
  Serial.begin(115200);

  Serial.println();
  Serial.print("Connecting to ");
  Serial.print(ssid);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.println();
  //Serial.println("WiFi connected");
  //Serial.print("IP address: ");
  Serial.println("Wifi Connected Success!");
  Serial.print("NodeMCU IP Address : ");
  Serial.println(WiFi.localIP());
```

Appendix F: Source code for Wi-Fi connection establishment and pin initialization