Faculty of Electrical and Electronic Engineering Technology

DEVELOPMENT OF AN IOT BASED SMART PET CAGE BY USING A MICROCONTROLLER

AZIM BIN ZAINAL LUDIN

Bachelor of Electronics Engineering Technology (Industrial Electronics) with Honours

2023

DEVELOPMENT OF AN IOT BASED SMART PET CAGE BY USING A MICROCONTROLLER

AZIM BIN ZAINAL LUDIN

A project report submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Electronics Engineering Technology (Industrial Electronics) with Honours

Faculty of Electrical and Electronic Engineering Technology

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2023

DECLARATION

I declare that this project report entitled "Development of IoT Based Smart Pet Cage By Using A Microcontroller" is the result of my own research except as cited in the references. The project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : Azim

Student Name : AZIM BIN ZAINAL LUDIN

Date : 14.1.2023

APPROVAL

I hereby declare that I have checked this project report and in my opinion, this project report is adequate in terms of scope and quality for the award of the degree of Bachelor of Electronics Engineering Technology (Industrial Electronics) with Honours.

Signature : fauz

Supervisor name : Ir. Ts. Dr Mohd Fauzi Bin Ab Rahman

Date : 27/1/2023

Signature :

Co-Supervisor :

Name (if any)

Date :

DEDICATION

This thesis is dedicated to Zainal Ludin Bin Mohamad and Rubi Binti Ahmad, my beloved parents for their constant love, encouragement, and inspiration. To my supervisor Encik Ir. Ts. Dr Mohd Fauzi Bin Ab Rahman who never giving up to taught and guide me to complete my project. To my helpful classmate and hoursemate always keep supporting me.

ABSTRACT

As the number of people who own pets grows yearly, so makes the demand for higher-quality pet care products. This has pushed the Internet of Things (IoT) technology forward in this industry. Pet owners can use IoT technology to follow their pets' activities and whereabouts from afar, check their pets' health and even interact with them. All of these smart pet care gadgets are becoming increasingly important in the lives of pet owners. We use IoT technology in this project to create an integrated system that includes a pet food feeder, water dispenser, and litter box, which are the three most important features pet owners will be concerned about when they are busy or away from their pets, especially cats or dogs. With NodeMCU Esp8266, the three subsystems are connected to the local network. Furthermore, the information gathered by each sensor is processed and displayed on a smartphone app. Thus, pet owners can access all information about their pet's food and water consumption. Furthermore, the owner will be able to monitor their pet using a camera that will be installed inside the smart pet cage and connected directly to the apps. Additionally, the application has a controlling function that allows pet owners to dispense food and water anytime and from any location. This project will provide pet owners with an efficient, convenient, and low-cost tool for pet care by incorporating this project's pet care system into a smartphone application.

ABSTRAK

Oleh kerana jumlah orang yang memiliki haiwan peliharaan bertambah setiap tahun, begitu juga permintaan untuk produk penjagaan haiwan kesayangan yang berkualiti tinggi. Ini telah mendorong teknologi Internet of Things (IoT) maju dalam industri ini. Pemilik haiwan peliharaan boleh menggunakan teknologi IoT untuk mengikuti aktiviti haiwan kesayangan mereka dan dari jauh, memeriksa kesihatan haiwan kesayangan mereka dan bahkan berinteraksi dengan mereka. Semua alat penjagaan haiwan kesayangan ini menjadi semakin penting dalam kehidupan pemilik haiwan kesayangan. Terdapat penggunaan teknologi IoT dalam projek ini untuk membuat sistem terpadu yang meliputi dispenser makanan haiwan peliharaan, dispenser air, dan kotak pasir, yang merupakan tiga ciri terpenting. Dengan NodeMCU Esp8266, ketiga subsistem disambungkan ke rangkaian tempatan. Selanjutnya, maklumat yang dikumpulkan oleh setiap sensor diproses dan dipaparkan pada aplikasi telefon pintar. Oleh itu, pemilik haiwan peliharaan dapat mengakses semua maklumat mengenai penggunaan makanan dan air haiwan kesayangan mereka. Selanjutnya, pemilik akan dapat memantau haiwan kesayangan mereka menggunakan kamera yang akan dipasang di dalam sangkar haiwan kesayangan pintar dan disambungkan terus dengan aplikasi. Selain itu, aplikasi ini mempunyai fungsi pengendalian yang membolehkan pemilik haiwan peliharaan membuang makanan dan air pada bila-bila masa dan dari mana-mana lokasi. Projek ini akan menyediakan pemilik haiwan peliharaan dengan alat yang cekap, mudah, dan murah untuk penjagaan haiwan kesayangan dengan memasukkan sistem penjagaan haiwan kesayangan projek ini ke dalam aplikasi telefon pintar.

ACKNOWLEDGEMENT

First and foremost, I would like to express my gratitude to my supervisor, Ir. Ts. Dr Mohd Fauzi Bin Ab Rahman for his precious guidance, words of wisdom and patient throughout this project.

I am also indebted to Universiti Teknikal Malaysia Melaka (UTeM) for the financial support throughout the project development which enables me to accomplish the project. Not forgetting my fellow colleague, housemates for the willingness of sharing his thoughts and ideas regarding the project.

My highest appreciation goes to my parents and family members for their love and prayer during the period of my study. Finally, I would like to thank all the staffs at the Universiti Teknikal Malaysia Melaka (UTeM), fellow colleagues and classmates, the faculty members, as well as other individuals who are not listed here for being co-operative and helpful.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

LIST OF ABBREVIATIONS

V	-	Voltage
VCC	-	Voltage Common Collector
GND	-	Ground

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

This chapter focuses on making thought and understanding the review of the venture, detailing the objectives, concisely the problem statement and scope and providing the project's outcome. In this way, the structure of the entire undertaking can be decisively imagined.

1.1 Background

People nowadays tend to seek joy in their lives. Some of them will go for outings or holidays, shopping, and many more. Some people keep pets in their homes for their happiness. Ownership is both a joy and a challenge. Once enticed to retain pets, it is impossible to give up one's fondness for them, to the point where pets become a part of the family and a part of life and are impossible to abandon. It is a long-lasting love. According to an article on the Insight website, an online survey done by Rakuten Insight's proprietary panels shows that 59% of the population in Malaysia has pets in their homes. 20% of people in Malaysia have dogs as their pets. Meanwhile, 34% of people in Malaysia choose cats as their pets. This indicates that more than half of the Malaysian population has owned a pet. As a result, a system that focuses on pet care can enable individuals to spend more time doing other things.

1.2 Problem Statement

Nowadays, people tend to have companions such as pets in their houses to avoid being lonesome at their home or to protect their homes. Usually, people in Malaysia will have dogs or cats as their pets, and some of them have exotic pets such as lizards and snakes. These pets need extra care from their owner. Although those pets needing to be taken care of, sometimes the owner may have other things to do such as working, travelling or any other unexpected things to do. Usually, people will ask someone close to look for their pet. It is a traditional way to take care of their pets while they are away. However, it may not only create problems for others, but also not knowing how to care for your pet properly. As a result, some individuals may utilise a more contemporary method, such as a pet feeder that will feed their pet at a time set by the user or a system that will feed their pet automatically when they stand in a certain location.

Most of the time, pet food and water will be exposed to the environment, including dirt and vermin, which may cause contamination. This will result in disease, as well as a slew of other issues. The food or water will soon run out, putting their dogs at risk of famine. The litter box, as we know, needs to be cleaned regularly. Otherwise, it will attract pests and it can end up contracting fungal infections like ringworm or hookworm. And then, if you're in contact with an infected cat, it can easily pass to you.

Although there may be a system on the market that can take care of the pet independently, the pet owner will still need to purchase all of the supplies in order to take care of the pet more conveniently. They will need to acquire pet food, an automated pet door, a pet tracking system, and a monitoring system, for example. All of the various components will not only take up all of the available rooms in the home, but they will also make it more difficult for the user to put everything up on their own. Users must, for example, build their main entrance to allow their pet to pass through, locate someone to wire the camera to watch their pet, and so on. All of these will raise the cost of adopting a pet in the user's home. As a result, the smart pet cage may help the pet owner address his or her present issue. The all-in-one system will make it simpler for pet owners to care for their pets.

1.3 Project Objective

The purpose of the research is to apply some technologies to existing pet cages and make it able to be remotely monitored. Specifically, the objectives are as follows:

i) To develop a reliable automatic pet water and food dispensing system, and pet litter box controlling system

ii) To design an IOT interface that can monitor and control the pet’s food and water dispensing as well as the pet’s litter box cleaning system via wifi connection.

iii) To evaluate the reliability and performance of the designed smart pet cage system.

1.4 Scope of Project

To avoid any uncertainty in this project due to some limitations and constraints, the scope of the project is defined as follows:

i) Smart pet cage acts as a monitoring system for the owner that can be used remotely through smartphones.

ii) The usage of a camera in smart pet cage serves as a security system that allows the user to keep an eye on their pets at any time.

iii) Using the technology applied, smart pet cage can control the quality and quantity of food or water.

iv) The litterbox will be kept clean by the automated operation.

1.5 Contribution of Research

i) It has a monitoring system for user to monitor their pet’s condition remotely and has security features.

ii) This proposed system provides a moveable system which makes it reliable and easy to use.

iii) Some tasks can be completed by the user without difficulty and from a distance.

iv) The system can manage the needs of pets even though owners are not present.

1.6 Thesis Outline

Based on the objectives previously presented and on the approach proposed before, this thesis is made up of five (5) chapters, which contents are summarised as follows:

i) Chapter 1. Introduction. This chapter presents the background of the study, problem statement, project objectives, scope of the project, contribution of research and thesis outline.

ii) Chapter 2. Literature review. This chapter begins with a brief review of the smart pet cage's application. This chapter will also summarise and compare several prior projects with similar concepts to this one. It will also cover the project's features, characteristics, and technology.

iii) Chapter 3. Methodology. The methods employed in this project is described in this chapter. This project's flow is demonstrated to effectively achieve the purpose. In addition, the hardware used to construct the smart pet cage system is covered in the Hardware Development section, whilst the software and approach utilised in the monitoring system is discussed in the Software Development section.

iv) Chapter 4. Preliminary result. In this chapter, the developed models are put to the test in terms of system performance and accuracy in real time. The system's functions are being tested to ensure that the desired outcome is achieved.

v) Chapter 5. Conclusion. This chapter summarises the main findings and successes of the research undertaken for this study, as well as areas for future research.

CHAPTER 2

LITERATURE REVIEW

This chapter focuses on the researcher, journal, article's expertise and philosophy, past analysis and technique comparisons, preliminary analysis and approach comparisons.

2.1 Introduction

In this modern's day, physical monitoring of living things is now possible with just a click of a mobile application interface thanks to the Internet of Things (IoT). It enabled devices to connect and communicate with each other via the Internet with relatively little human intervention.Various systems have been used for both living and non-living topics, and the Internet of Things (IOT) is constantly growing.

2.2 Previous related works

This project includes new features that improve on prior projects and provide additional benefits. With the support of an existing project, a comparison may be conducted by selecting the best of the best and learning from their mistakes in order to produce a superior result. Each project used different components and methods, but the end result was the same: pet cage upgrade. Furthermore, a greater grasp of this existing project was required in order to eliminate needless repetition.

2.2.1 PetCare: A Smart Pet Care IoT Mobile Application

According to [1] Physical monitoring of living things is now possible with just a click of a mobile application interface thanks to the Internet of Things (IoT). It enabled devices to connect and communicate with each other via the Internet with relatively little human intervention. The project involves implementing the Internet of Things (IoT) to create a smart pet care assistant in a mobile application.

Based on research done by [1], the project is about implementing the Internet of Things (IoT) to create a smart pet care assistant in a mobile application. The goal of the project is to create an embedded system for monitoring and caring for domestic pets. The project entails remote meal scheduling, camera service monitoring, room temperature and light sensing, smart pet door control, excrement pad cleaning, and remote music and voice command activation at home while pet owners are at work or engaged in outdoor activities.

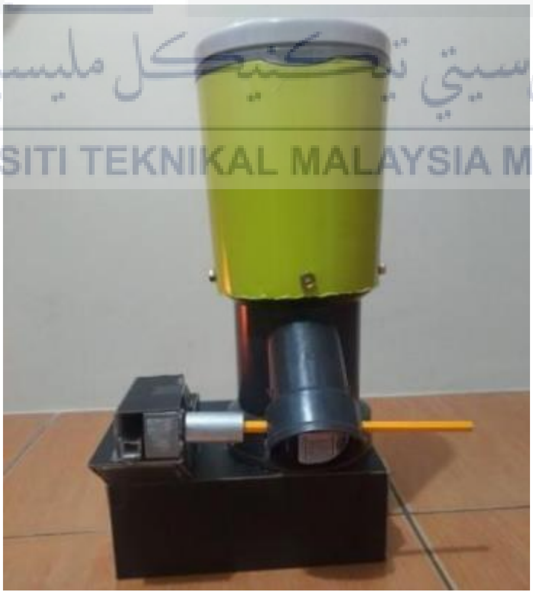


Figure 2.1 Food Dispenser Design Mock Up

The pet's consumption is handled by an automated feeding system, which includes time schedules and the proper amount of food to serve. As shown in Figure 2.1, is a automated feeding machine that was built to feed the pets. The smartphone application provides control, and both the feeding system and the mobile app are connected to the internet. The rotation of the dispenser fins for the automated food with time schedules is controlled by a DC Gear Motor attached to the system. This idea includes a smart door system that allows the pet to go in and out of the pet door while also preventing other pets from entering the residence. The sensor is affixed to the pet's collar, and the door detects when the pet is around. This project also includes a monitoring system. When the user is away from home, he or she has real-time access to view pet behaviour. It can be viewed using the camera embedded in the smartphone application.

Plastic Water Solenoid Valves are used to control water flow. The automated feeder includes both solid food and water to keep the pet hydrated. It only runs in one direction and requires very little water pressure. The project includes a Door Locking System that utilises an FS90 Micro Servo as a revolving lock for the smart pet door. It is linked to a proximity sensor, and when it detects the pet collar, which is linked to another proximity sensor, the servo unlocks the pet door, allowing the pet to pass through.

2.2.2 Smart Cage Implementation with Dependable Safety Agent for Dogs

K. S. Ng et al, [2] developed a smart cage for a dog is shown in the research, which includes an automatic excrement detection module, a cleaning module, a feeding module, and several more sensors. Excrement is identified in the smart cage using image processing technologies, and the conveyance cleans it up automatically. Furthermore, with its built-in and adjustable speed feature designed for different levels of training, the conveyance may also be used as a treadmill for the dog to exercise. However, because the smart cage is fully automated, the pet owner's engagement is minimal.

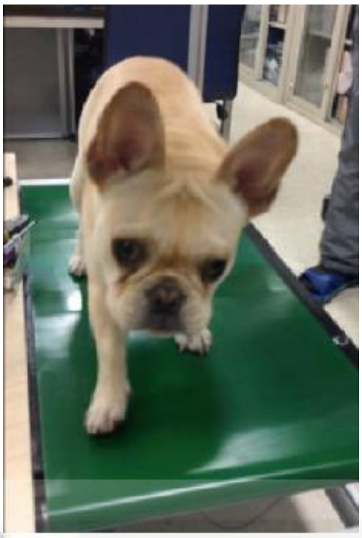


Figure 2.2 The Conveyance module

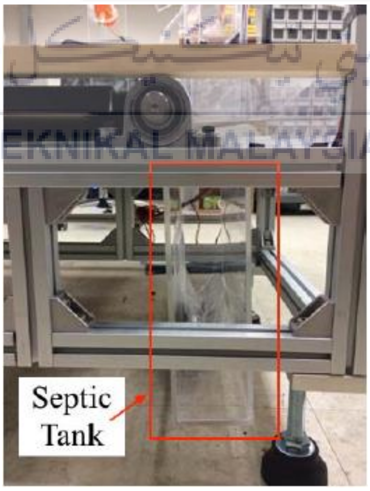


Figure 2.3 The Septic Tank

While the conveyance is still in operation as shown in Figure 2.2, a dependable safety agent will be offered to address the safety concern. The trustworthy safety agent will keep an eye on two types of data before deciding whether or not to halt the conveyance rotation. The first type is an image processing system that detects the movement of the dog in the cage and signals the conveyance to come to a complete stop if the dog moves unnaturally. The second type of information utilised to halt the conveyance is by using a current sensor to identify anomalous current peaks in the conveyance.

As shown in Figure 2.3, a septic tank was added to contain the faeces that pets will produce. An energy-saving system and a smart feeding system are included in this project. The energy-saving technology employs a data-mining technique to determine the best frame rate for detecting cameras. It has the potential to reduce the camera's unneeded power consumption. The smart feeding device provides forage based on the moisture content of the dogs' faeces. The moisture content of the dog’s excrement can be utilised as a quantitative measure of their food.

2.2.3 Smart Cat House with Telegram-based Control System

Based on recent project’s research [3], through the social media of the owner's telegram, cat owners may monitor and regulate schedules for eating, temperature, and drinking conditions in a cat cage with this Smart Cat Home. The Raspberry Pi 3 was used to create Smart Cat Home. This system is used to monitor and give feed schedules based on cat habits, as well as to know the weight of cat feed detected by a loadcell, monitor water willingness to drink using a water level sensor, capture cat conditions via web cameras, and monitor temperature using the DHT11 sensor. The Telegram bot will display all Smart Cat Home system conditions.

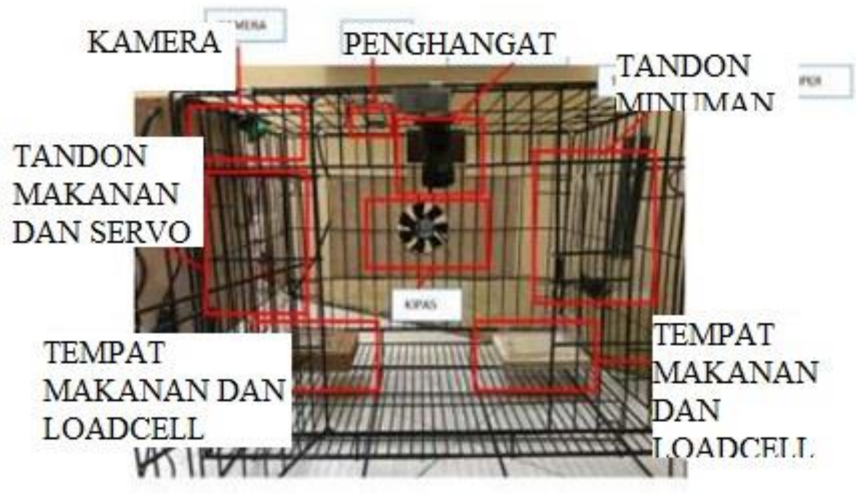


Figure 2.4 Actual Smart Cat Home System

Smart Cat Home's entire system can function properly. As shown in Figure 2.4, all component are installed inside the cage. The exhasut fan function when the temperature drops below 22 degrees Celsius, the warming system kicks in, and when the temperature rises beyond 26 degrees Celsius, the fan kicks in. The weight sensor in the eating system can correctly read the weight of the feed. An average of 78 grammes of feed was determined from 5 data points on feed weight. On the drinking system, the drinking water reservoir will fill the drinking container when the water level sensor detects a voltage of 1600 mV, indicating that the water level is 2.5 cm below the drinking container's surface. It is known that there is a 33.8 ms delay in data transmission from the calculation of the delay value. The Smart Cat Home’s is a great system which will give a lot advantages to the user. It help to feed the pets automatically without hustle the pet’s owner.

2.2.4 MiAU: Aid System for Dogs Care

P. P. Cruz [4] has developed a system to assist pet owners with pet care. A dog care system called MiAU has been proposed, investigated, and designed in order to unify and connect existing items. MiAU's fundamental purpose in design is to be an integrated dog care system that can suit the needs of all sorts of dog owners, whether they are experienced or not. Its goal will be to not only bring all of the different types of modules together on a single platform to which both owners and doctors will have access, but also to develop a product that is inexpensive for owners. As shown in the Figure 2.5, all the module are gathered in a single container.

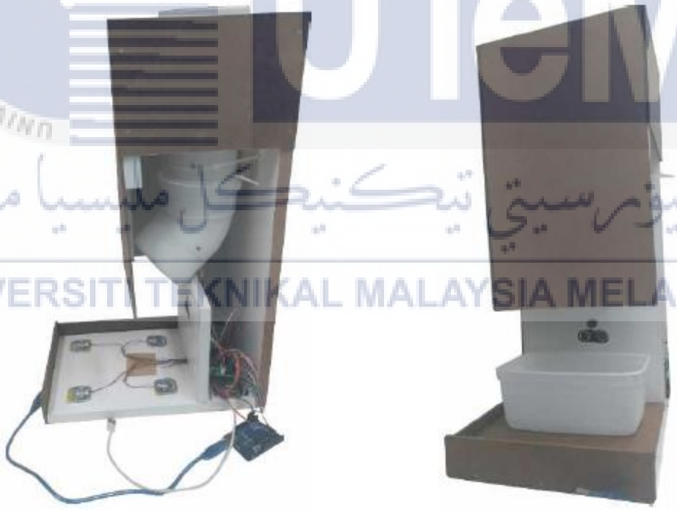


Figure 2.5 Feeding System Prototype

A dog care system called MiAU has been developed, investigated, and designed, consisting of an intelligent collar which is completely dedicated to monitoring the health of the pet, a smart feeder and a water dispenser using its mobile app, pet owners can set up automatic feeding schedules and portion sizes for their pets and a water dispenser as well as a mobile application. A preliminary investigation of a biometric system for dog pawprint recognition was also included in the project. A veterinarian and a pet store assisted in the design and development of the entire system.

2.2.5 Smart Bran Dispenser for Animal Husbandry Purpose

Smart Bran Dispenser for animal husbandry purpose is a project developed by Muhammad Alif Azanuddin Alias et al [5]. The goal of this project is to develop an innovative animal husbandry management system to increase the efficiency of the agricultural system, notably in terms of livestock nutrition and feed supplies.The other goal is to make it easier for breeders to feed their animals, which can be done remotely using a tracking module that sends a signal to the user and informs them of the bran dispenser's status via the Blynk server. The primary controllers were implemented using NodeMCU ESP8266 and Arduino UNO.

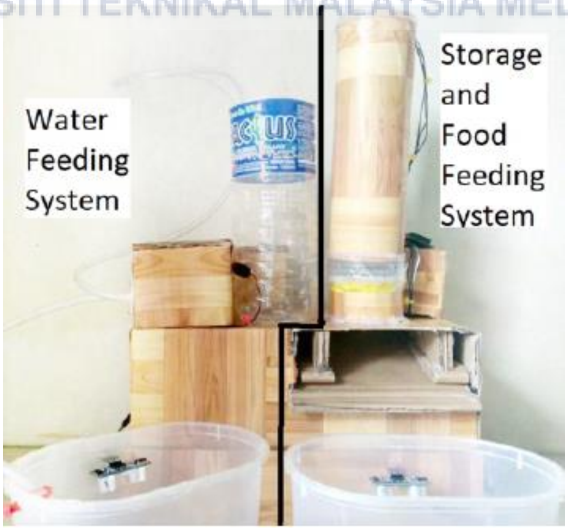


Figure 2.6 Smart Bran Dispenser Prototype

In this project, there are water feeding system and food feeding system as shown in Figure 2.6 which connect to a WiFi system based on the ESP8266 that allow breeders to regulate the bran dispenser so that it feeds at the appropriate moment. It includes a GPS locating device, which allows the Blynk server to solve the problem of the food feeder's stock position and determine whether food is available in the storage box. After collecting these values, the computer performs the activities necessary to operate the solenoid valve. The breeder can use the GPS to find out which stock has run out of food. Because the entire system is autonomously controlled by the Internet of Things system, this technology helps the breeder reduce the cost of hiring people (IoT). In this project, a smart bran dispenser prototype is created utilising a NodeMCU ESP8266 and a Blynk server to provide the location of the food dispenser system.

2.2.6 Arduino Mega based Pet Feeding Automation

B. Ravi Babu, P. Pavan Kumar and Dr. P. G. Kuppusamy [6] developed an Arduino-based pet feeding automation that can automatically feed your pet at the appropriate moment. A gravity feeder is the most basic type of pet feeder, consisting of a hopper full of food that falls into a bowl as the pet empties it. The authors propose to utilise the DS3231 RTC Module to display the time on an LCD To provide the food, a servo motor rotates the containers. It allows you to customise the rotation angle and container opening duration based on the amount of food you wish to feed your pet. To manually set the time for feeding the Pet, use the matrix keypad. The DS3231 RTC (Real Time Clock) Module was used to programme the time and date when the owner's pet should be fed. As a result, the device drops or fills the food bowl automatically based on the owner's pet's eating routine. The Arduino Ethernet Shield allows users to connect Arduino to the internet quickly and easily. This shield allows the Arduino to send and receive data from anywhere in the world via the internet. It is in charge of connecting the mobile device to the Arduino Mega board.

Automatic pet feeders are designed to deliver a certain amount of food at specific intervals, ensuring that your pet is fed on time. Many pets must be fed at various times throughout the day. The automatic food dispensers work by setting the time on the appliance's internal timer, eliminating the need for an external clock. Meanwhile, the food is kept in a sealed container to prevent spoilage until the owner decides and the device releases it. As shown in Figure 2.7, all actuator are connected to the microcontroller which is Arduino Mega.

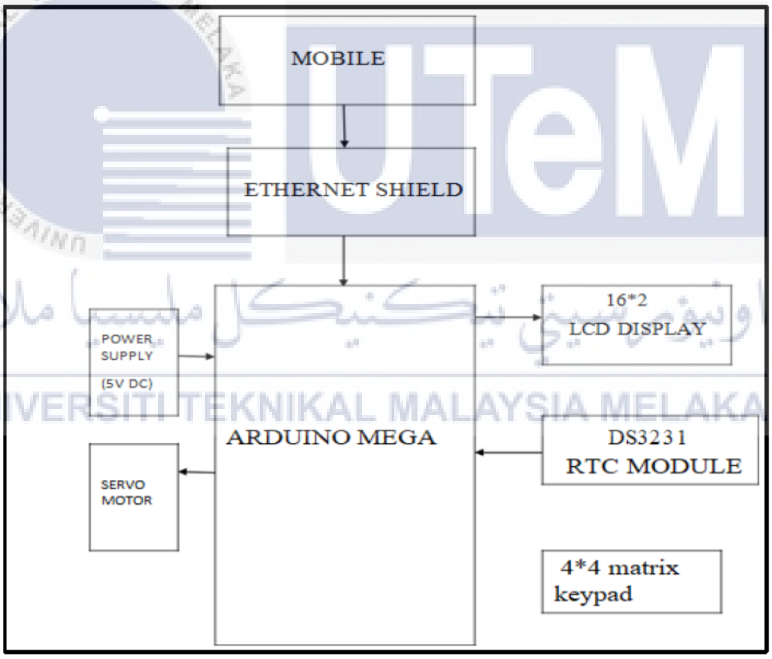


Figure 2.7 System Block Diagram

2.2.7 Smart Pet Feeder

According to [7] , the authors stated that The Smart Pet Feeder is an autonomous feeding equipment for pets that works with apps on a smartphone. The gadget has a closed video circuit that allows users to observe their pets when they are away from home, and it is designed to assist in the feeding process. The machine is named after an existing notion that has been improved by the addition of an internet connection. The Arduino UNO Rev3 and the servo motor are among the machine's novel components, which may last for a week with electrical connections.

It may be used automatically, allowing users to stop worrying about their pets. Those who care about animals will undoubtedly choose this machine for their pets at home. Consumers will also be less concerned about leaving their pets for extended periods of time, such as when returning home or working full-time. Finally, especially in the mechanical industry, it is an excellent technique to enhance and utilise top and local users. Author also state in [7], The Smart Pet Feeder is quite useful in petting procedures as well as boosting the local industries.

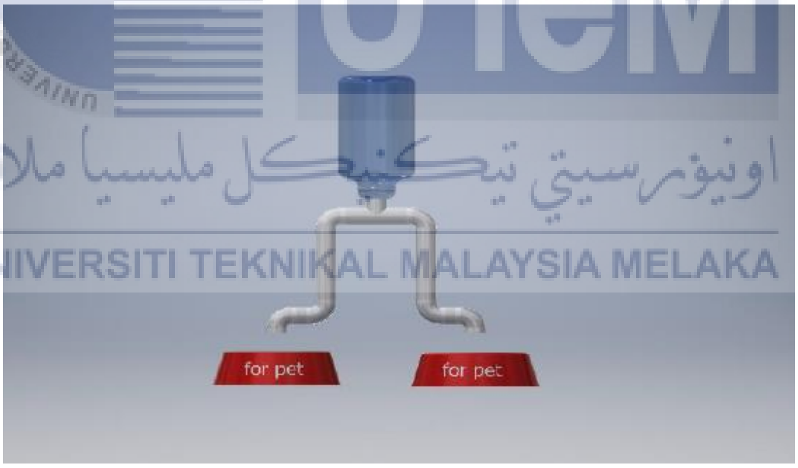


Figure 2.8 Proposed Design

2.2.8 Automatic Cat Feeder And Location Tracker

Automatic cat feeder and location tracker is a project that have been developed by Nur Izzatul Nadia Binti M Razif *et al*, [8]. The authors designed an automatic cat feeder with a position tracker. The Arduino UNO and Node MCU ESP8266 stand-alone modules, combined with appropriate sensor types such as Ultrasonic Sensor, Servo Motor, and GPS module tracking position, are unique. This device was created to feed pets when their owners are away from their houses.

The owner can utilise the mobile application to request that the bowl be refilled by pressing a button on the system. In addition, the device notifies the owner when the food and water levels in the bowls reach critical levels. Finally, by tracking the position of the cat, it can assist pet owners in being attentive. The research resulted in a completely functional prototype that can be interacted with remotely using a mobile application. This results in less labour, more leisure, and fewer worry for the individual.

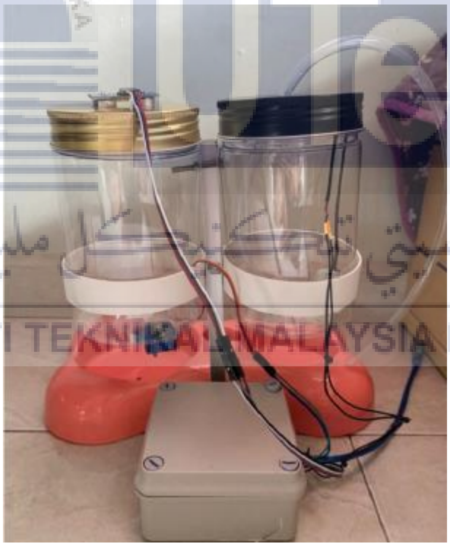


Figure 2.9 System Prototype

The system works with a few sets of sensors that are placed near where the cat eats and drinks, as shown in Figure 2.9. The Blynk Application is used to link all of the sensors to the system. After the timer on Blynk is set, the machine begins to dispense. The sensor sends the data wirelessly to the Arduino UNO, which uses the Arduino IDE to record the activities.

2.2.9 Automatic Cat Litter Box

Nurul Shuhada Binti Muhamad Shukor *et al*. [9] has developed a system that helps pet owner in maintaining cleanliness of pet cage called automatic cat litter box. The goal of this project is to save cat owners time by inventing a cat litter box that can work automatically, as well as to improve cat safety while cleaning. Recognize that there is a minor issue with manually used cat litter boxes, namely that cat owners do not have enough time to clean and dispose of faeces due to their daily jobs, the faeces disposal work is inefficient, and the safety of cats during cleaning operations for other automatic cat litter boxes is quite hazardous. The fabrication process began with an assessment of the issues and wants of cat owners in dealing with this product.

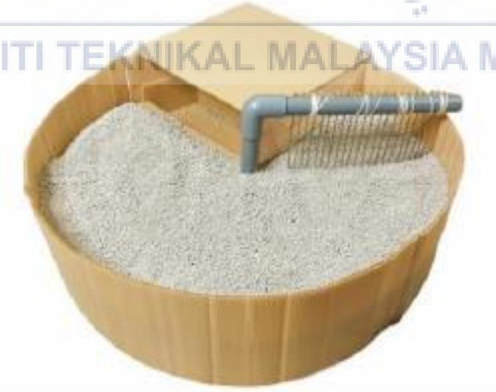


Figure 2.10 Actual Automatic Cat Litter Box

The system's design is suited for pets such as cats, dogs, and small animals, as shown in Figure 2.10. When the system detects faeces, it will automatically toss it away. This device will assist owners in keeping their pet's litter box clean by using it.

2.2.10 Smart Pet’s Litter Box

According to [10], Smart Pet's Litter Box is a project that aims to help pet owners solve the most common issue they have when cleaning their pet's litter box. The major goal of this project is to build a Smart Pet Litter Box that will assist pet owners in dealing with their pet's faeces in a simple and practical manner by utilising an automated system that combines mechanical and electronic devices. This project detects the presence of pets using a PIR motion sensor device as an input signal and delivers the data to an Arduino PIC microcontroller. During the cleaning process, the computer sends commands to the stepper and servo motors, which move the rack and operate the waste tray.

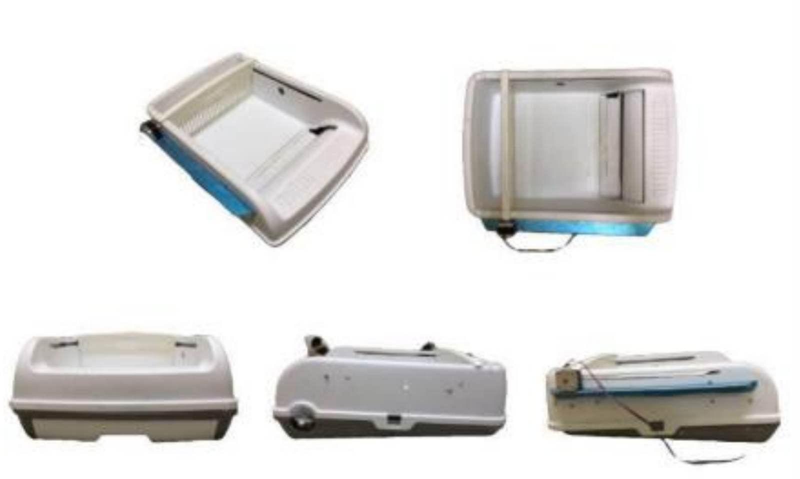


Figure 2.11 Prototype of the main body

This prototype (Figure 2.11) was a success, and the cleaning procedure ran well, with an average cleaning time of 3 minutes per process and a clean percentage of 75%. In terms of suggestions, this project might be enhanced by incorporating a noise suppressor for stepper motors, which would reduce motor noise. Finally, as a result of this project, pet owners will no longer be burdened with cleaning, and they will be able to save time, money, and energy while keeping the greatest degree of cleanliness in their homes and surrounds.

2.3 Summary of previous related projects

Table 2.1 Summary of Previous Related Projects

No.	Related projects		Objectives	Method/Component used	Advantages
	Title	Author			
1.	PetCare: A Smart Pet Care IoT Mobile Application	A. A. A. Luayon [1]	to create an embedded system for monitoring and caring for domestic pets	• Raspberry Pi Model B+ • DC Gear Motor • Plastic Water Solenoid Valve • Camera	User able to monitor their pets remotely by using designated app.
2.	Smart cage implementation with dependable safety agent for dogs	K. S. Ng [2]	develop an embedded system for house pet monitoring and care.	• capacitive sensor • camera	Detect the movement of the dog in the cage, with the purpose to signal for the conveyance to stop immediately if the dog moves abnormally.
3.	Smart Cat House with Telegram-based Control System	A. S. A. Putri [3]	To monitor and regulate schedules for eating, temperature, and drinking conditions in a cat cage.	• DHT11 sensor • Raspberry Pi 3 • Web cam Logitech C170 • Servo motor • Fan	User able to monitor the condition of the cage and pets by using mobile phone application.
4.	MiAU: Aid System for Dogs Care	P. P. Cruz [4]	proposes a way of collecting the dataset using a common camera, a	• Stepper motor • Ultrasonic Sensor • NodeMCU	User able to monitor the condition of cage and location of

			way to segment the dataset images and a way to classify them using a K-NN algorithm.	• HX711 ADC Converter	the pets by using mobile apps. The apps stored data about the pets.
5.	Smart Bran Dispenser for Animal Husbandry Purpose	M. A. A. Alias [5]	to develop an innovative animal husbandry management system to increase the efficiency of the agricultural system, notably in terms of livestock nutrition and feed supplies.	• Arduino UNO • NodeMCU Esp8266 • GPS module • Ultrasonic sensor	helps the breeder locate the bran dispenser's location from somewhere that runs out of bran in the storage and details the longitude and latitude through the Geolocation API.
6.	Arduino Mega based Pet Feeding Automation	T. A. Rao [6]	to ensure the time to time feeding of pet in absence of its master so that master can do his other tasks without worrying about feeding.	• Arduino Mega • DS3231 RTC Module • Servo motor • Ethernet shield	Pets will be supply with fresh food and with right portion. The food will be stored in sealed container that keep the food from spoil.
7.	Smart Pet Feeder	S. Danial [7]	No longer need to worry about their pets during business trips or vacations and allows users to input different pet's profiles into the pet feeder via Android phones.	• Arduino UNO Rev3 • Servo motor • D-link IP Camera	SMART PET FEEDER has high security for home use and very useful to monitor the food quantity of pet food.
8.	Automatic Cat Feeder And Location Tracker	N. Izzatul [8]	helps the pet owner to ensure their beloved cats are being fed in good condition mostly when the cat owner is unable to feed their cat directly at home.	• Arduino UNO • NodeMCU Esp8266 • GSM module • Ultrasonic sensor	The GPS cat tracking collar are simply a tracking device that attached to cat by tracking the location of cat in real time and user can use mobile apps to control the food dispenser.
9.	Automatic Cat Litter Box	N. Syuhada [9]	Designing a cat litter box that can operates	• Solar panel • Arduino Uno • DC motor	This project saved time and energy to clean up cat

			automatically and to help users save their time to dispose cat’s waste everyday.		waste.
10.	Smart Pet’s Litter Box	S. Nuraini [10]	To design a Smart Pet’s Litter Box to help pet owners deals with their pet’s excrement in an easy and practical way.	• PIR motion sensor • Arduino UNO • Servo motor	It helps pet owners save time and effort because the litter box cleans itself after each use, pet owners won’t have to clean it or scoop up its faeces manually

2.4 Summary

This chapter has described and compared previous projects that are most similar to this one. Many data from prior projects, such as title, software and hardware used, approach, and others, were collected.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter will explain the processes and methodology used in this project. This chapter will also discuss the hardware and software implementation in this project. The flow chart of the proposed system is also presented. Aside from that, component specifics and software are addressed following a proper selection to aid project development.

3.2 Project Methodology

The project technique is illustrated in Figure 3.1. Preliminary investigation, information analysis, and identification, decision making, software and hardware development and analysis, and testing of developed projects are the five stages of the proposed project approach. The goal of developing this project methodology is to ensure that everything runs properly during the project.

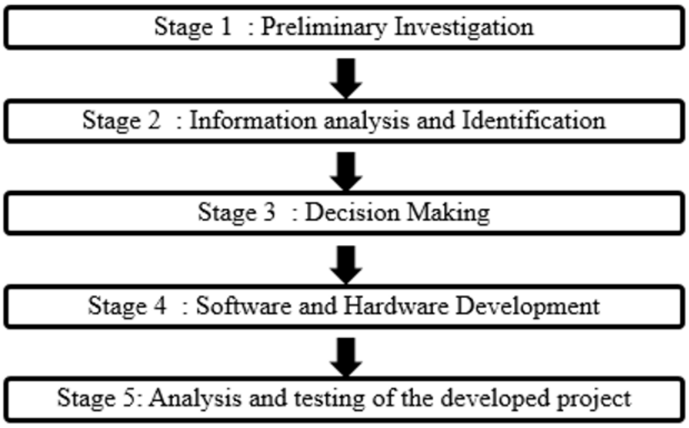

Stage 1 : Preliminary Investigation
Stage 2 : Information analysis and Identification
Stage 3 : Decision Making
Stage 4 : Software and Hardware Development
Stage 5: Analysis and testing of the developed project

Figure 3.1 Project Stage Flow

The purpose of preliminary investigation is to learn more about project title choosing. To gain a better understanding of the project, research will be conducted on the chosen project title. The problem statement and project objective can be derived from the preliminary investigation's findings. In stage 2, the information analysis and identification stage, researchers combed through existing studies to see what method or technique was employed in this paper. The method and methodology will be listed so that they can be compared. In the Literature Review section, this material will be documented. The selection of hardware and software in stage 3 decision-making should be based on the study conducted in stage 2. After having a better understanding of the comparison of the approach and technique employed in the previous article, the study in stage 2 makes choosing in stage 3 easier.

Following the selection of hardware and software, project development planning must be completed in order for the project to move smoothly. The system can be developed at stage 4, which includes software and hardware development. Stage 4's outcome should be in line with the paper's goal. Stage 5 will consist of analysing and testing the generated project, as well as validating the results. This can be accomplished in a variety of methods, including analysing raw data from the project or comparing the results to those of a prior article.

3.3 Project Process Flow

To show all of the steps involved in completing a project, a process flow figure have been done. The project's process flow is shown in Figure 3.2, which begins with the user action on the mobile application that connect with the project’s system.

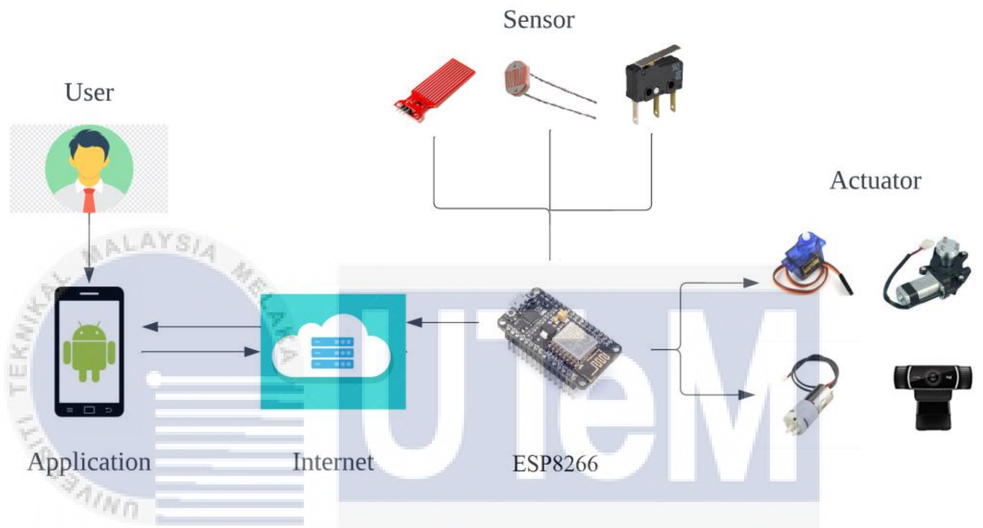


Figure 3.2 Process Flow Diagram

As shown in Figure 3.2, the system begins with the user turn on the application in mobile phone. After that, the apps will be connected to the system through internet. Then, in the case of data obtained from mobile apps, it is up to the user selection which there will be few options for the user to choose which each option will represent the existing features in this project. Those features will be monitoring system, pet feeding, water dispensing, door lock system and litter box cleaning system. All these features are connected to the Arduino through wi-fi module and internet. As for sensor, it will give notification to the user about the data collected through mobile apps. Once user notify, they can control the actuator using the button inside the mobile apps.

3.4 Block Diagram of the Project

It is required to construct a block diagram to demonstrate the overall structure of the project. The flow chart and the block diagram are similar. Input and output frameworks are the primary emphasis of block diagrams. Figure 3.3 shows the components that are connected to microcontroller in the system. There are few sensors such as water level sensor, LDR and limit switch which it will serve as a sensor in the circuit. Water level sensor are used to measure the water level inside the tank and LDR will serve as detector for the food container. Water pump, power window motor, servo motor and camera will be the actuator for this project. All the components will be connected to the NodeMCU Esp8266 as microcontroller. NodeMCU will control the actuators according to the input that it received

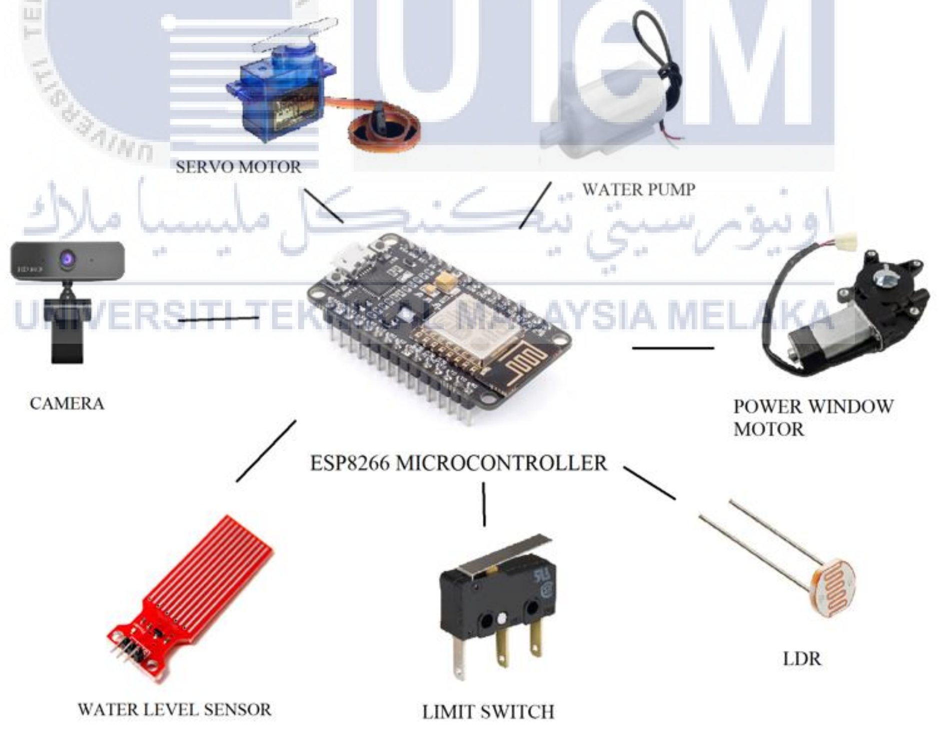


Figure 3.3 Component Mind Map

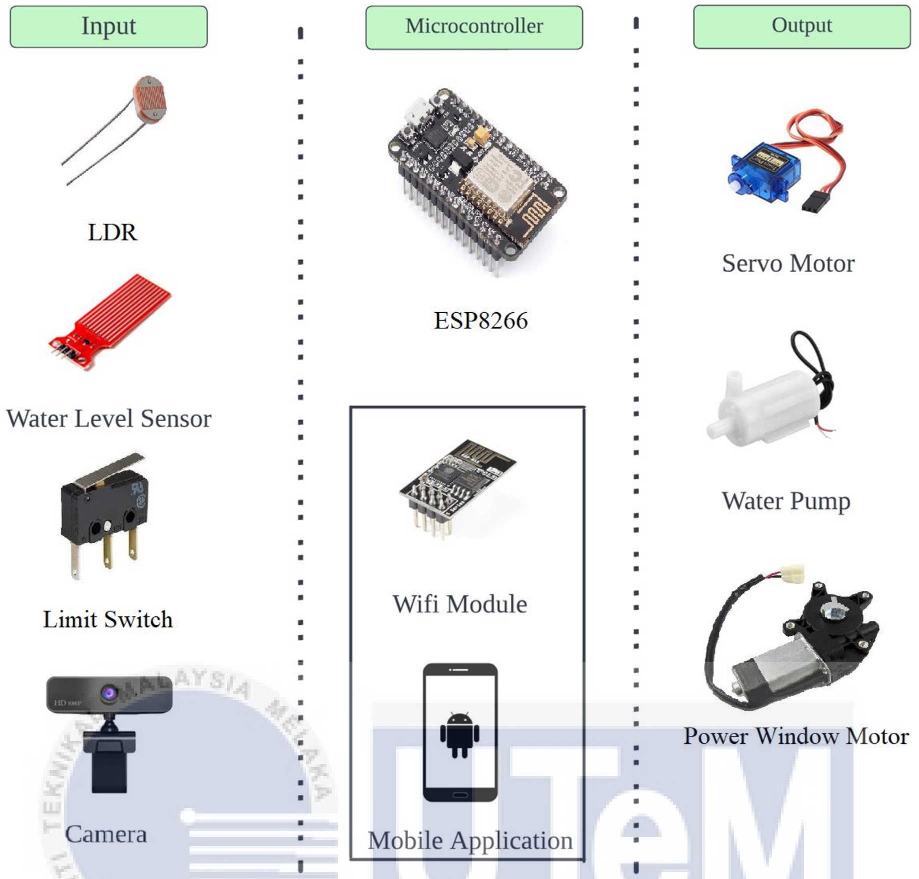


Figure 3.4 Project Block Diagram

The input, microcontroller, and output components of the block diagram shown in Figure 3.4 are divided into three categories. From the beginning to the finish of the process, each segment represents the components and objects involved. This project will only work if the target object, which is the pets within the cage, is there. When motion is detected, it sends a high signal to the I/O pin and sends the data to NodeMCU Esp8266, which activates the wi-fi module chip and sends a notification update to the owner's connected device. The owner will be notified through notification. From this point forward, the owner will have complete authority over the system and will be able to issue commands to it.

As shown in the Figure 3.4, the wi-fi module and mobile application are put together in a box. It is because this two are connected to each other for software development. When it comes to software NodeMCU Esp8266 reads the input and converts it into output by passing encoding instructions to the microcontroller. The wifi features inside the ESP8266, will allow the user to connect the circuit to their phone.

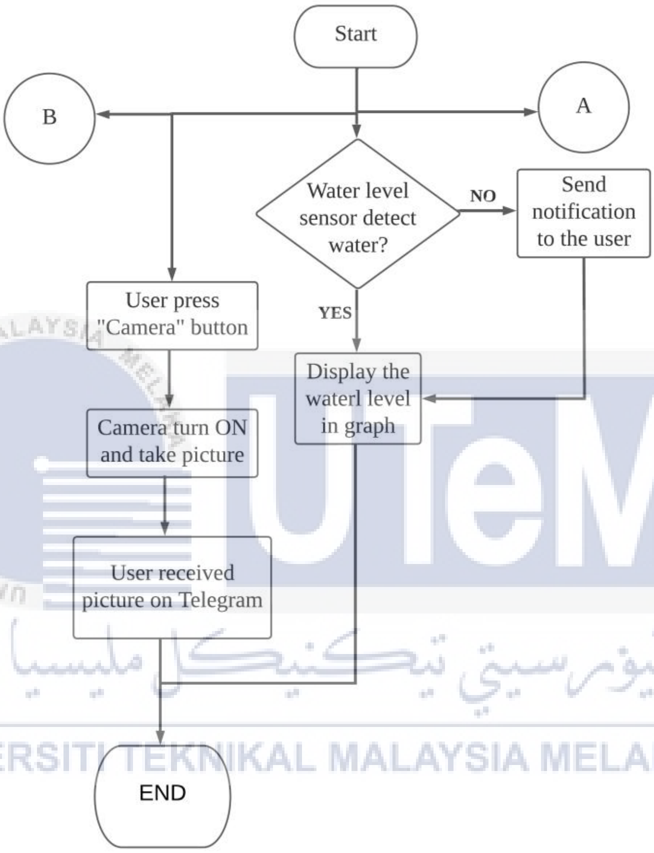

Start
B
A
Water level sensor detect water?
NO
Send notification to the user
YES
User press "Camera" button
Display the waterl level in graph
Camera turn ON and take picture
User received picture on Telegram
END

Figure 3.5 Project Flowchart

Camera is utilized to capture the current situation of the cage once the user clicks on monitoring features on the mobile application as shown in Figure 3.5. The picture taken will be send to the user’s Telegram apps on their mobile phone. As for water level sensor, it works by detecting the volume of water inside the water tank. It will generate graph on Blynk apps once it detected the volume of water inside water tank. If there is no water inside the water tank, it will send notification to the user.

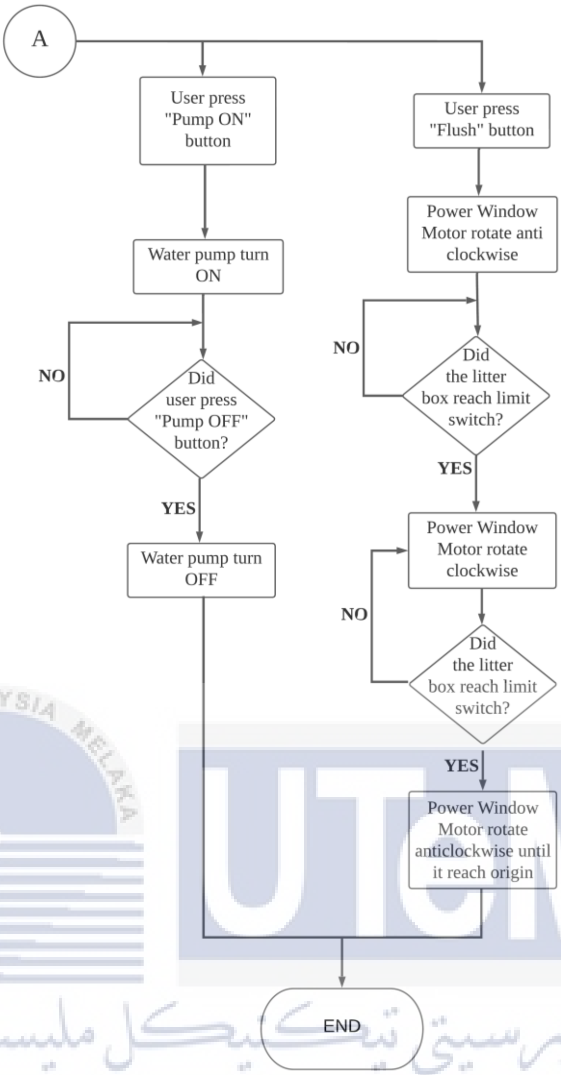

A
User press "Pump ON" button
Water pump turn ON
NO
Did user press "Pump OFF" button?
YES
Water pump turn OFF
User press "Flush" button
Power Window Motor rotate anti clockwise
NO
Did the litter box reach limit switch?
YES
Power Window Motor rotate clockwise
NO
Did the litter box reach limit switch?
YES
Power Window Motor rotate anticlockwise until it reach origin
END

Figure 3.6 Project Flowchart

In Figure 3.6, the water dispenser system will start to work once the owner or user press the "Pump ON" button in the application which will trigger the water pump to turn on and flow water into the water bowl. The water pump will turn off once user pressed "Pump OFF" button and if the user did not press the button, the water pump will keep turning on. The litter box is related to the usage of power window motor which it will cause the litter box to rotate upward and downward. The condition of the litter box will be controlled by user through mobile application.

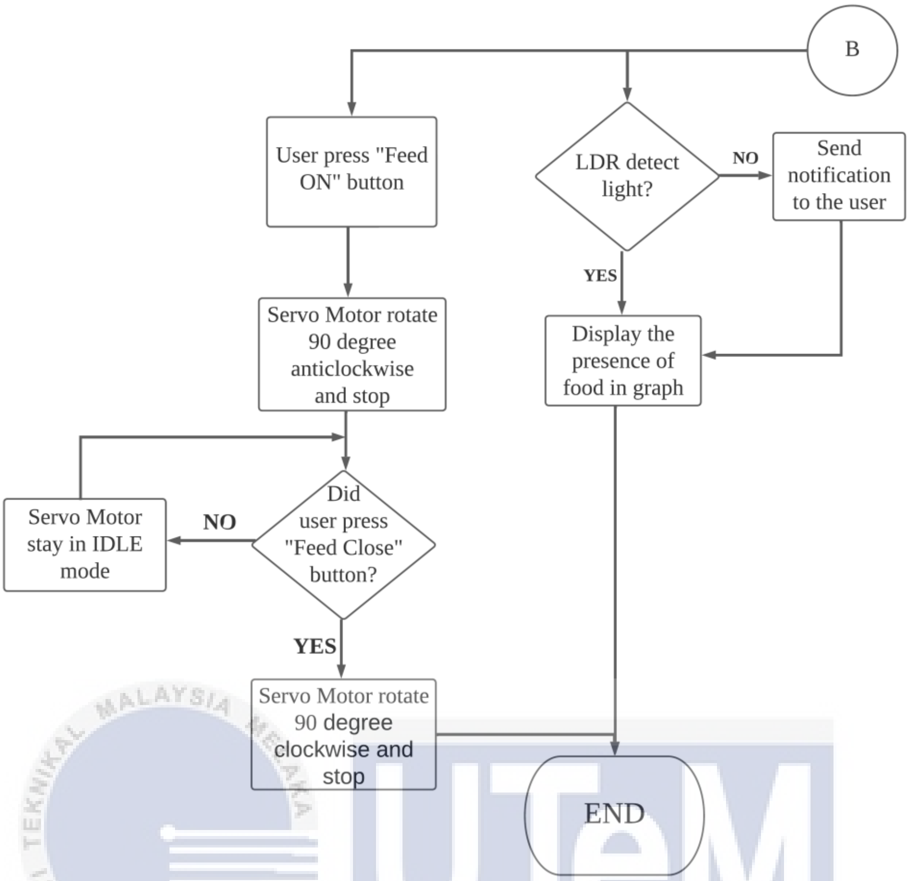

B
User press "Feed ON" button
LDR detect light?
NO
Send notification to the user
YES
Servo Motor rotate 90 degree anticlockwise and stop
Display the presence of food in graph
Did user press "Feed Close" button?
NO
Servo Motor stay in IDLE mode
YES
Servo Motor rotate 90 degree clockwise and stop
END

Figure 3.7 Project Flowchart

For feeding system, the system will start to work once the owner or user press the feed button in the application which will trigger the servo motor to drop pet feed onto the bowl as shown in Figure 3.7. The LDR sensor will act as indicator to the food that will notify user through the mobile app. Then it will send data to the microcontroller which is NodeMCU Esp8266 and send notification to the mobile application. The litter box is related to the usage of power window motor which it will cause the litter box to rotate upward and downward. The condition of the litter box will be controlled by user through mobile application.

3.5 Hardware Implementation

3.5.1 Circuit Design

This project consist of few sub system which all connected by internet. Those system are pet feeder system, water dispenser, litter box and door lock system. All this system has it own component and circuit connection.

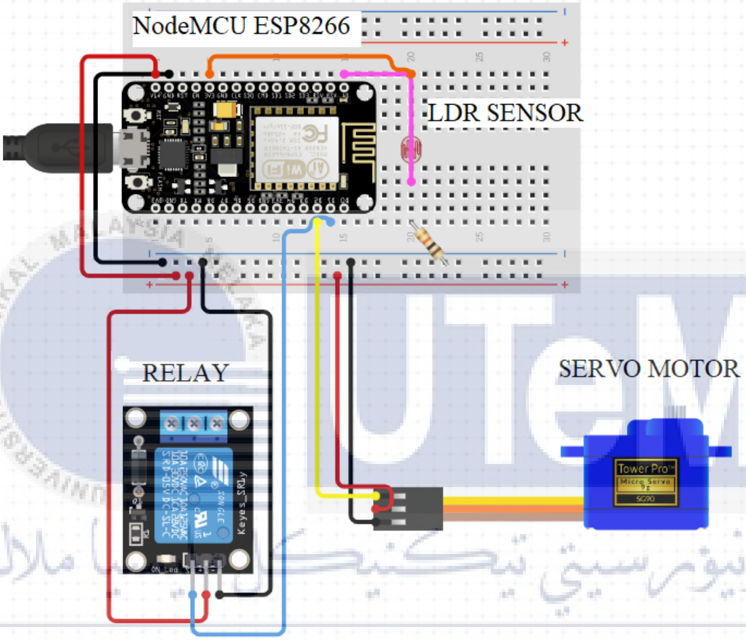


Figure 3.8 Pet Feeding System Design

LDR sensor and servo motor will be used in the pet feeder system. Figure 3.8 depicts their relationship. In order for a LDR sensor to work, it requires two input pins to be connected. It does not have polarity, which is positive and negative. It will connect to the NodeMCU Esp8266 digital input. It also has a servo motor, whose output pin connects to the NodeMCU Esp8266 digital input. The LDR sensor will detect light intensity inside the food container and transmit a signal to NodeMCU Esp8266. It will notify the user via a mobile application of the amount of food within the container as soon as it detects excessive light intensity. Once the user presses the feed button on the mobile application, the servo motor only starts to work. The primary purpose of the relay module is to turn on and off the servo motor.

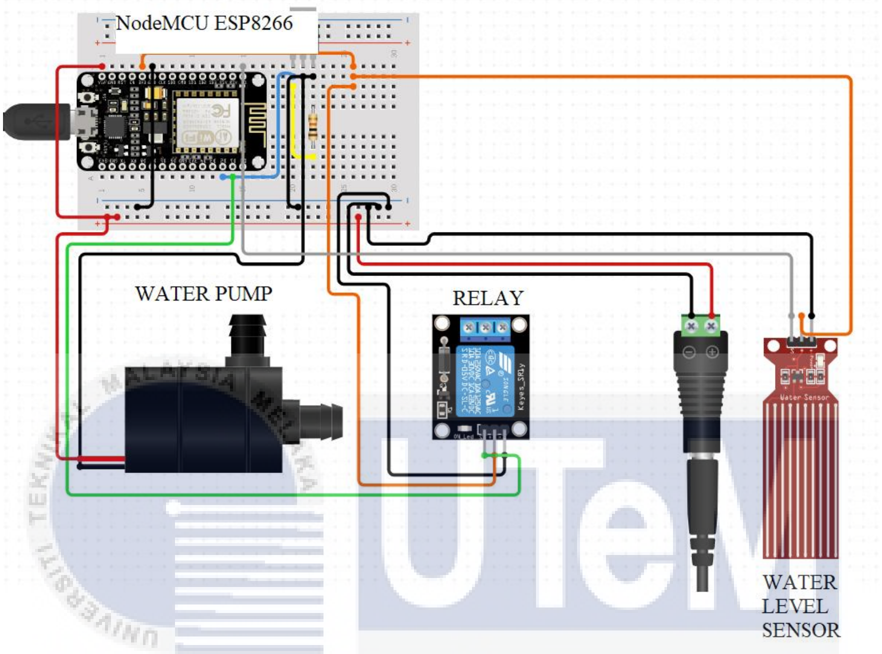


Figure 3.9 Water Dispenser Circuit Design

According to the connection in the Figure 3.9, water level sensor and water are used, indicating that the system is a water dispenser. The NodeMCU Esp8266 digital input pin is connected to the water level sensor. When the water level is low, the digital output increases, and the sensitivity can be adjusted. It simply need 3.3V to 5V to function. When the digital output goes high, it sends a signal to NodeMCU, which will notify the user. This project's water pump is a 5V Mini Water Pump that is submersible. The flow rate is between 1.2 and 1.6 litres per minute. Because the flow rate of this pump is very modest, it is easy to control. Only when the user presses the pump button on the mobile apps does the water pump operate.

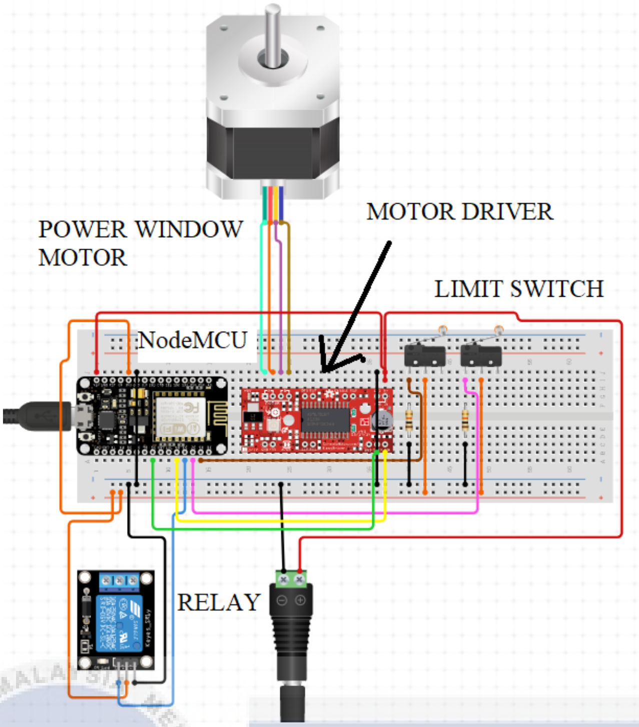


Figure 3.10 Litter Box System Design

In Figure 3.10, the automated litter box switch has two limit switches as sensors, one will cause the motor to go forward, while the other will cause the motor to rotate backward. Normally Open (NO) or Normally Close (NC) are the two conditions for the limit switch's three pins that connect to the NodeMCU. The Power Window Motor, which serves as the actuator in the litter box system, will be connected to the NodeMCU via a motor driver as a controller. The power window motor has a rated speed of 60 RPM and can run on 12 volts. It boasts 2.9 N.m of torque as well. Once the user presses the button flush on the mobile application, it will send signal to the NodeMCU and the power window motor will only start to operate. It will revolve initially backward until it reaches a limit switch, then forward until it reaches a second limit switch before returning to the starting position.

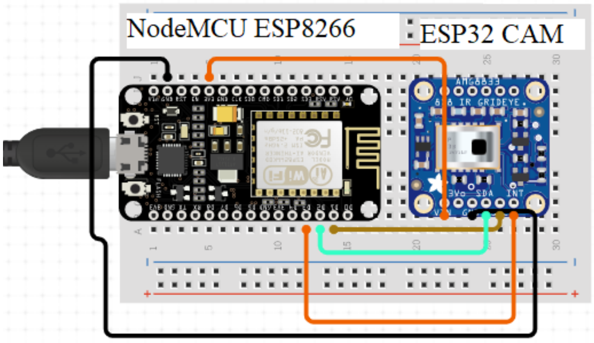


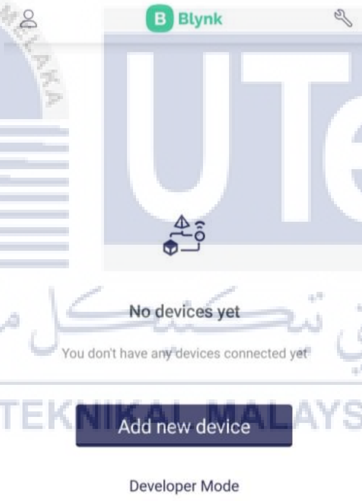
Figure 3.11 Monitoring System

As shown in Figure 3.11, there is a microcontroller and a camera. In this monitoring system, it only have one actuator which is ESP32-Camera. The ESP32-CAM is a small size, low power consumption camera module. It can run on just 5 volts. There are only three pins from the ESP32-Camera that need to be connected to the NodeMCU: Vin, ground, and TX pin. The user will be able to keep an eye on their pets thanks to the Esp32, which will take a photo of the present scenario inside the cage. The user merely needs to activate the camera in the mobile application by clicking the "Camera ON" button. It will then capture the current situation inside the cage after sending a signal to the NodeMCU Esp8266. The ESP32-Camera will produce an image that will be sent to the user's Telegram application.

3.6 Software Implementation

3.6.1 Blynk Application

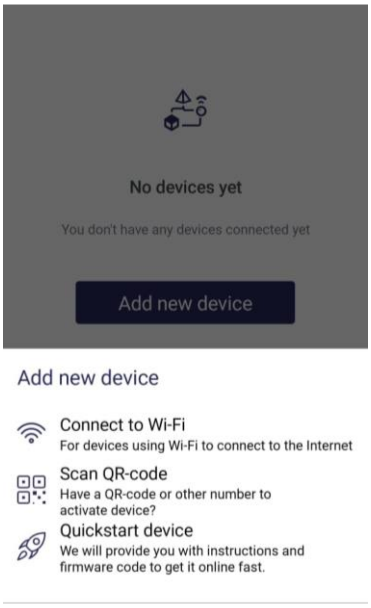
Blynk is an iOS and Android platform that allows you to control Arduino, Raspberry Pi, and other similar devices over the Internet. It is a digital dashboard where you may drag and drop widgets to design a project's graphical interface. Everything is simple to put together. Blynk is not tied to a specific board or shield. Instead, it works with whatever hardware installed. Whether Arduino or Raspberry Pi is connected to the Internet by Wi-Fi, Ethernet, or this new ESP32 chip, Blynk will get online and ready for the Internet of Things.



No devices yet

You don't have any devices connected yet

Figure 3.8 Blynk Home Page



No devices yet

You don't have any devices connected yet

Add new device

Add new device

Connect to Wi-Fi
For devices using Wi-Fi to connect to the Internet

Scan QR-code
Have a QR-code or other number to activate device?

Quickstart device
We will provide you with instructions and firmware code to get it online fast.

Figure 3.9 New Device Instruction

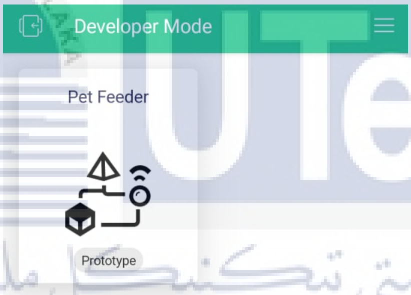


Figure 3.10 Developer Mode Home Page

It is the home page for the Blynk application, as illustrated in Figure 3.8. To build a new application, the user must first click Add New Device, which will display a list of options for finding a new device, as shown in Figure 3.9. The ESP8266, a wi-fi module that connects to the internet, will be used for this project. The option "connect to Wi-Fi" must be selected because it will link the entire machine to the internet. When both programmes are connected, Developer Mode appears, as illustrated in Figure 3.10, and the application construction process can begin.

3.7 Gantt Chart

A Gantt chart is required to ensure that our plan runs successfully. It serves as a guideline to guarantee that the project is completed on time. The Gantt chart of project planning BDP 1 is shown in the table 3.1.

Table 3.1 Gantt Chart PSM 1

Tasks	W 1	W 2	W 3	W 4	W 5	W 6	W 7	W 8		W 9	W 10	W 11	W 12	W 13	W 14
PSM 1 briefing	🟩	🟩							**Mid Term Break**						
Study the component used		🟩	🟩												
Chapter 1				🟩	🟩										
Study previous related project					🟩	🟩	🟩								
Chapter 2					🟩	🟩	🟩	🟩							
Study mobile application development							🟩	🟩							
Hardware purchases								🟩		🟩					
Chapter 3								🟩		🟩	🟩				
Preliminary result											🟩	🟩			
Chapter 4												🟩	🟩		
First draft submission												🟩			
Second draft submission													🟩		
Report submission via ePSM														🟩	
Presentation															🟩

Tasks	W 1	W 2	W 3	W 4	W 5	W 6	W 7	W 8	Mid Term Break	W 9	W 10	W 11	W 12	W 13	W 14
PSM 2 briefing	🟦	🟦													
Circuit construct	🟦	🟦	🟦	🟦	🟦										
Chapter 3				🟦	🟦										
Apps development					🟦	🟦	🟦								
Chapter 4					🟦	🟦	🟦	🟦							
Prototype development							🟦	🟦							
Hardware installation								🟦		🟦					
Chapter 4								🟦		🟦	🟦				
Project Testing										🟦	🟦	🟦	🟦		
Chapter 5												🟦	🟦	🟦	
First draft submission												🟦			
Second draft submission													🟦		
Report submission via ePSM														🟦	
Presentation															🟦

3.8 Summary

This project finished if all phases are completed and all of the aforementioned conditions, including software and hardware, are met. Any issues that develop throughout the project's execution will be addressed in order to achieve the goals.

CHAPTER 4

RESULTS

4.1 Introduction

The goal of this project is to to develop a reliable automatic pet water and food dispensing system, and pet litter box controlling system. This chapter will go over the design and hardware implementation of the smart pet cage system. The performance analysis results are observed when the system does not function as planned. Motion sensor, servo motor, and Blynk app notification will all be put to the test as a consequence.

4.2 Data Analysis

In order to obtain information or facts that aid in decision-making, one method is to do a data analysis. It has a variety of strategies and methods to help get the desired outcome. To analyse the trends in each piece of data gathered from the project's construction, a frequent test was run. At the end of the process, a successful project can be created with superior data analysis obtained from the appropriate data analysis technique and tools.

4.2.1 Value of Light Intensity

The LDR sensor was used to measure light intensity and adjust settings in accordance with the environment. The sensor works by detecting the intensity of light inside the food container to measure the amount of food inside the container. When high light intensity is detected, it denotes that there is only a tiny amount of food still in the container, and it sends a signal to NodeMCU Esp8266 to notify the user. Pet food will be placed in the food container until it is completely filled. Once the container is filled with pet food, the LDR sensor will be completely covered on the bottom. As shown in Table 4.1 , there are data analysis collected from the LDR sensor. Mobile apps were used to measure the amount of light in the food container, and a multimeter was used to measure the voltage and resistance.

Table 4.1 Reading of light intensity, voltage and resistance for LDR sensor

Iluminance (lx)	Resistance (Ohm)	Voltage (V)
0	5000	2.4
101	3000	1.6
143	2000	1.4
193	1600	1.35
209	1400	1.25
268	1000	1.2

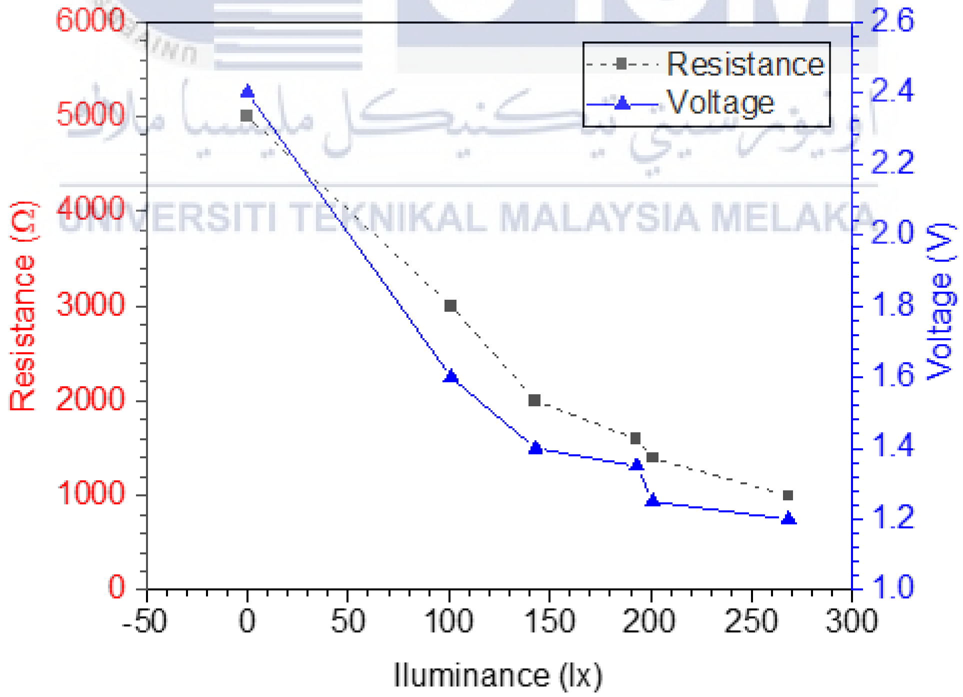


Figure 4.1 Graph of LDR Data Analysis

Resistance and voltage both rise in direct proportion. Voltage and resistance decrease as iluminance increases as shown in Figure 4.1. As resistance and voltage both rise, the graph rises. It demonstrates that the measured values of voltage and resistance decrease as light intensity by the LDR sensor increases.

4.2.2 Value of Water Level

The water level sensor was used to measure the water level inside the water dispesnser tank. The sensor works by detecting the resistance inside the food container to measure the water level. When the sensor was submerged under the water, it will measure the resistance and it sends a signal to NodeMCU Esp8266 to notify the user.

Table 4.2 Water level sensor data

	Volume (ml)	Voltage (V)
	0	0.46
	100	0.42
	200	0.38
	300	0.34
	400	0.28
	500	0.24

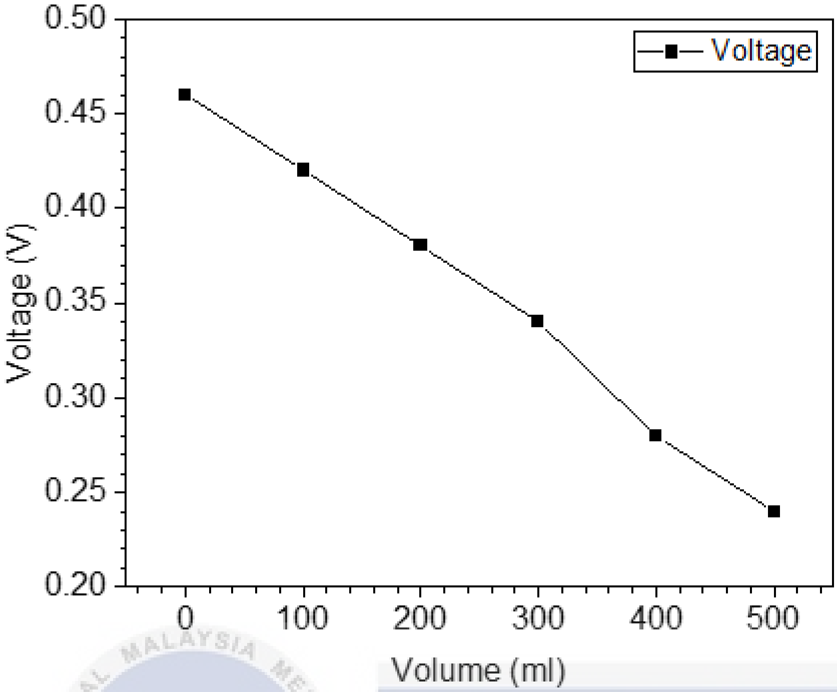


Figure 4.2 Graph of Water Level Sensor Data

Six volumes of water were tested to determine the voltage generated by the water level sensor based on the data obtained in Table 4.2. Figure 4.2 illustrates how the recorded voltage values decrease as volume rises. It demonstrates that the voltage detected increases with the depth of the water level sensor's submersion.

4.3 Result of Developing Process

4.3.1 Circuit Construction

In order to portray the connection of the rough idea of the circuit, an application named Multisim was used to sketch the circuit. Multisim is an open-source hardware that allow users to be creative as they could that is electronics accessible, especially to those who wants to experiment with their prototype. This circuit construction display an overview of all components used in this project that involved the input, process and output. As shown in

Figure 4.3, all component are connected to the microcontroller which is NodeMCU Esp8266. There are a few parts that Multisim does not include in their library. To mimic the actual component, output pins were used in place of the original components.

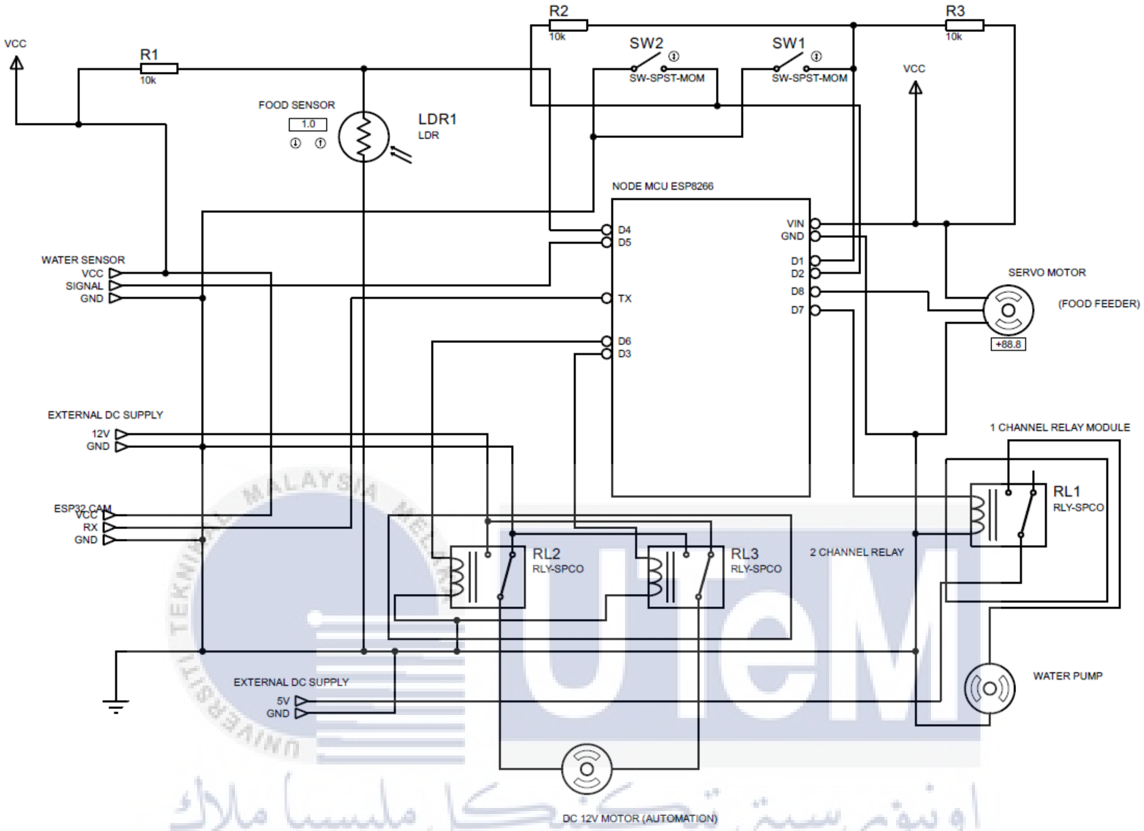


Figure 4.3 The overview of project circuit

4.3.2 Hardware Setup

For hardware setup, a circuit was constructed with cable interconnect each components. A 12V power adapter was used to supply the circuit in order for it to works. A system that can fit inside a cage was developed and shown in Figure 4.4. Figure 4.5 are the circuit that have been developed which shows that the NodeMCU Esp8266 as microncontroller and all the component are connected to it.

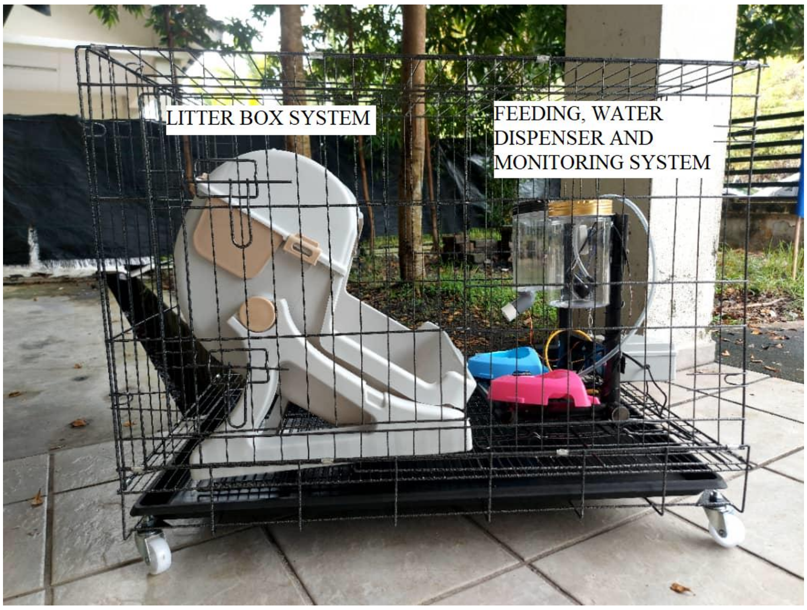


Figure 4.4 IoT Based Smart Pet Cage

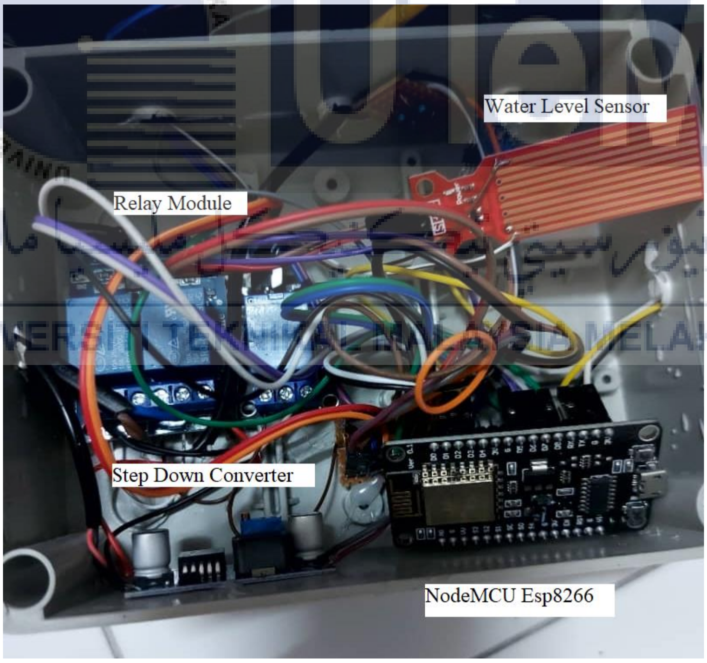


Figure 4.5 Project Circuit

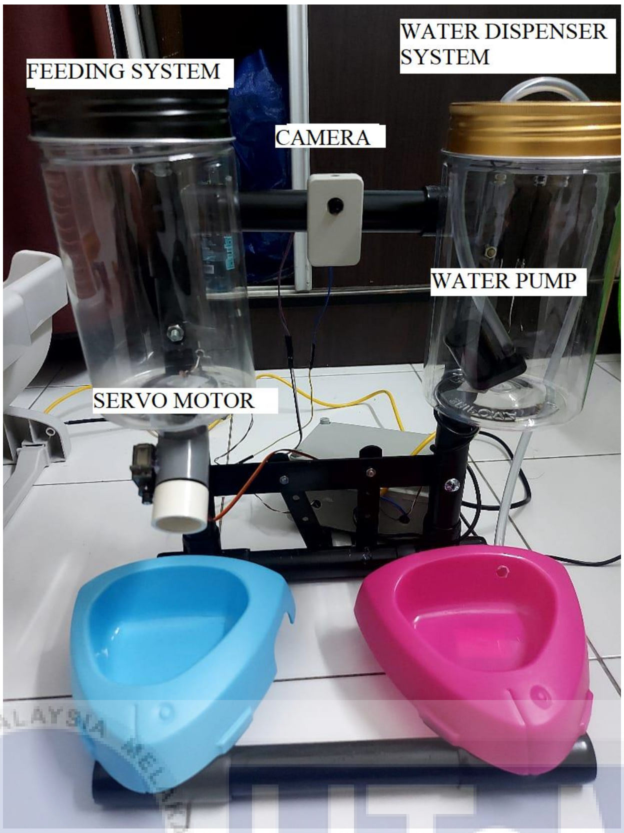


Figure 4.6 Feeding, Water Dispenser and Monitoring System

Three systems—a feeding system, a water dispenser, and a monitoring system—are installed to a PVC platform, as shown in Figure 4.6. Two bowls are present, one for food and the other for water. The food container has a servo motor placed at the base that opens up for the food to fall out. In order to transfer water from the water tank to the water bowl, a water pump was also added. Last but not least, a camera was placed in the middle for monitoring purposes to provide a better picture.

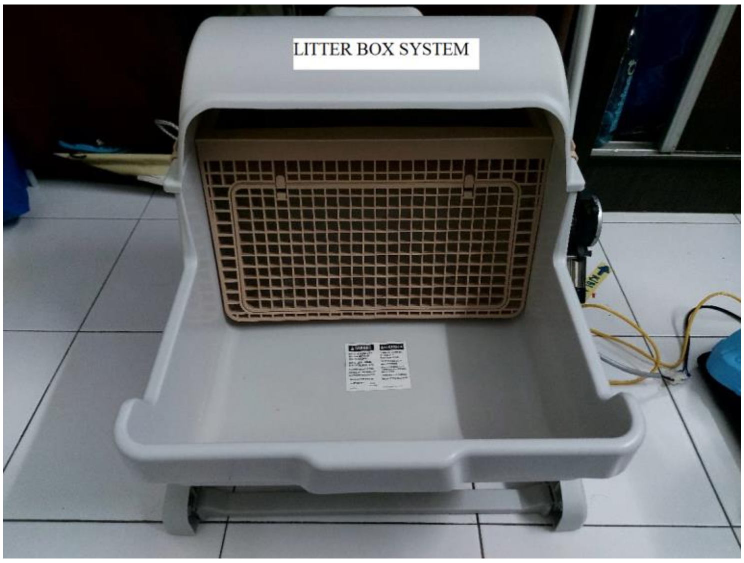


Figure 4.7 Litter Box System

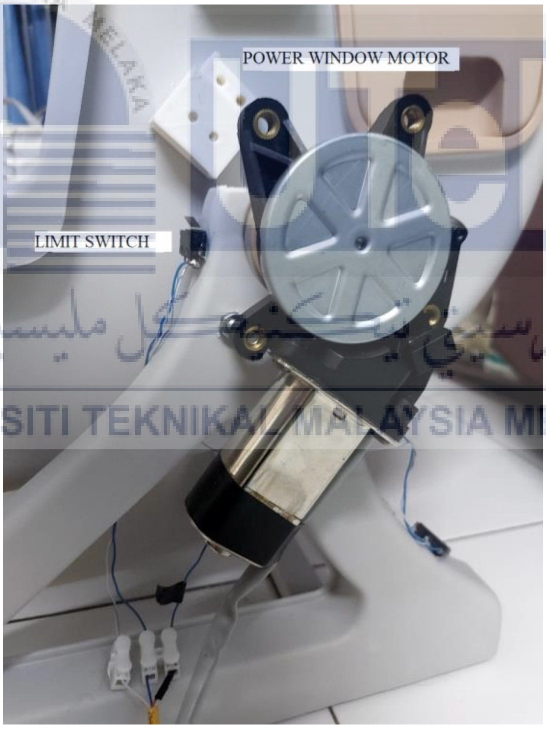


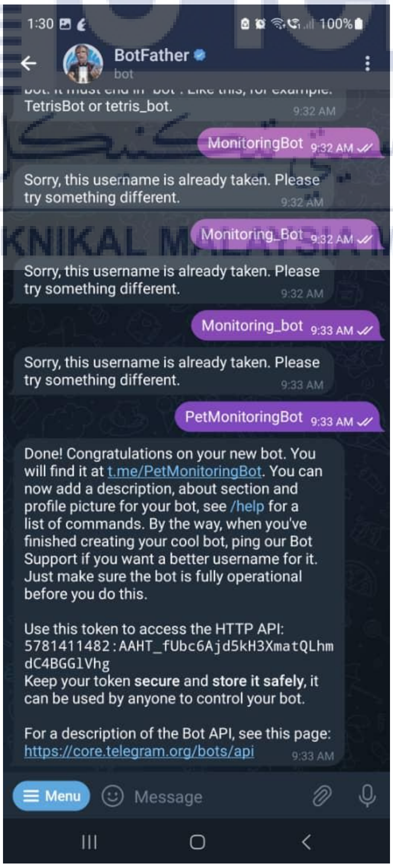
Figure 4.8 Litter Box Actuator and Sensor

In this project, an automated litter box was constructed, as seen in Figure 4.7. It includes two limit switches that serve as sensors to provide a signal for rotation and a power window motor. Figure 4.8 shows the limit switch and postition power window motor.

4.3.3 Software Setup

4.3.3.1 Telegram

Push notifications were employed as a method of notifying the user or the owner of the pet. The initiative used Telegram as a channel to convey the picture it had taken in order to alert users via their smartphones. As shown in Figure 4.9,with the aid of @BotFather, a new bot was developed to function inside of Telegram in order to make use of the Telegram feature. After receiving an allowed token to access the HTTP API, a new bot named @MonitoringBot was formed using a straightforward command setup in the @BotFather. To request a Chat ID from Telegram, the @getId bot was utilised. Tokens and Chat IDs obtained from the bot were both utilised in the coding for the camera setup as shown in Figure 4.10 .



bot. It must end in `bot`. Like this, for example: TetrisBot or tetris_bot. 9:32 AM

MonitoringBot 9:32 AM

Sorry, this username is already taken. Please try something different. 9:32 AM

Monitoring_Bot 9:32 AM

Sorry, this username is already taken. Please try something different. 9:32 AM

Monitoring_bot 9:33 AM

Sorry, this username is already taken. Please try something different. 9:33 AM

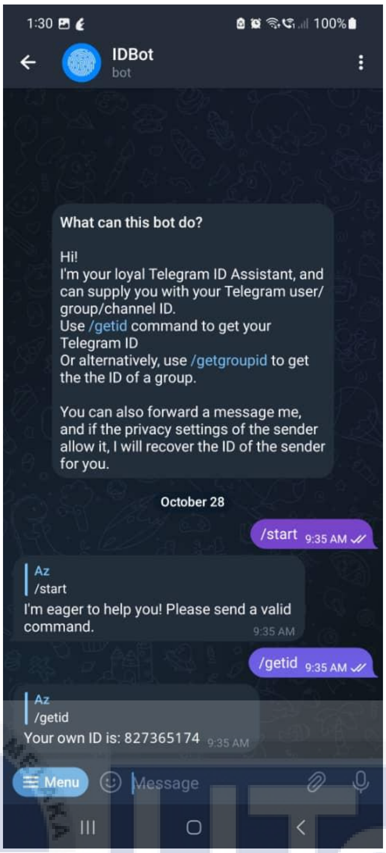
PetMonitoringBot 9:33 AM

Done! Congratulations on your new bot. You will find it at t.me/PetMonitoringBot. 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:
5781411482:AAHT_fUbc6Ajd5kH3XmatQLhmdC4BGG1Vhg
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 9:33 AM

Figure 4.9 BotFather Generator



What can this bot do?

Hi!
I'm your loyal Telegram ID Assistant, and can supply you with your Telegram user/ group/channel ID.
Use /getid command to get your Telegram ID
Or alternatively, use /getgroupid to get the the ID of a group.

You can also forward a message me, and if the privacy settings of the sender allow it, I will recover the ID of the sender for you.

October 28

/start 9:35 AM

Az
/start
I'm eager to help you! Please send a valid command. 9:35 AM

/getid 9:35 AM

Az
/getid
Your own ID is: 827365174 9:35 AM

Figure 4.10 IDBot Generator

Once Telegram is fully set up, the @MonitoringBot will begin receiving images over an internet connection as shown in Figure 4.11. The "Camera" button on the Blynk application will send a signal to the NodeMCU Esp8266, causing it to activate the ESP32-Cam and snap a picture of the current scene. Once the picture is taken, it will send directly to the @MonitoringBot in user’s Telegram just like in Figure 4.11

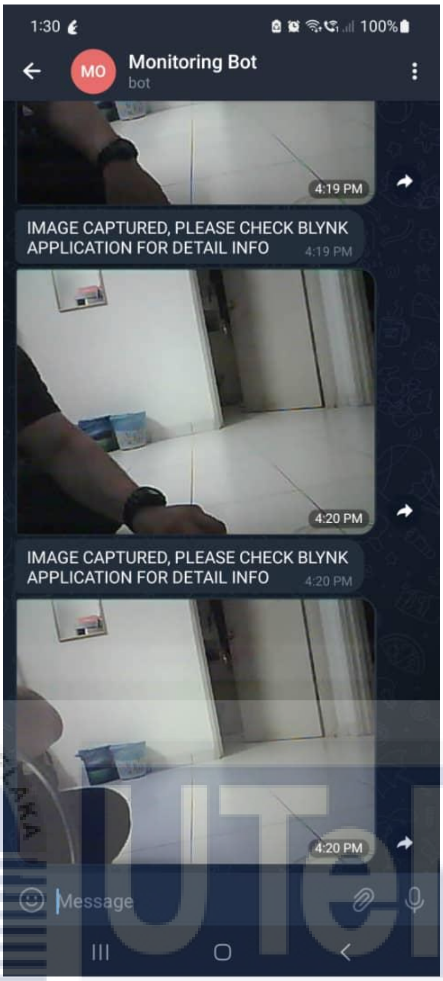


IMAGE CAPTURED, PLEASE CHECK BLYNK APPLICATION FOR DETAIL INFO 4:19 PM

IMAGE CAPTURED, PLEASE CHECK BLYNK APPLICATION FOR DETAIL INFO 4:20 PM

Figure 4.11 Monitoring Bot on Telegram

4.3.3.2 Blynk

Both Android and iOS devices can use Blynk. On all hardware and supported devices, the Blynk user interface and API are the same. Because of their ease of use and capacity to be managed via the virtual pin function, widgets are well-known. There are many ways for users to access the Blynk cloud service, including WiFi, BLE & Bluetooth, GSM, Ethernet, and USB.

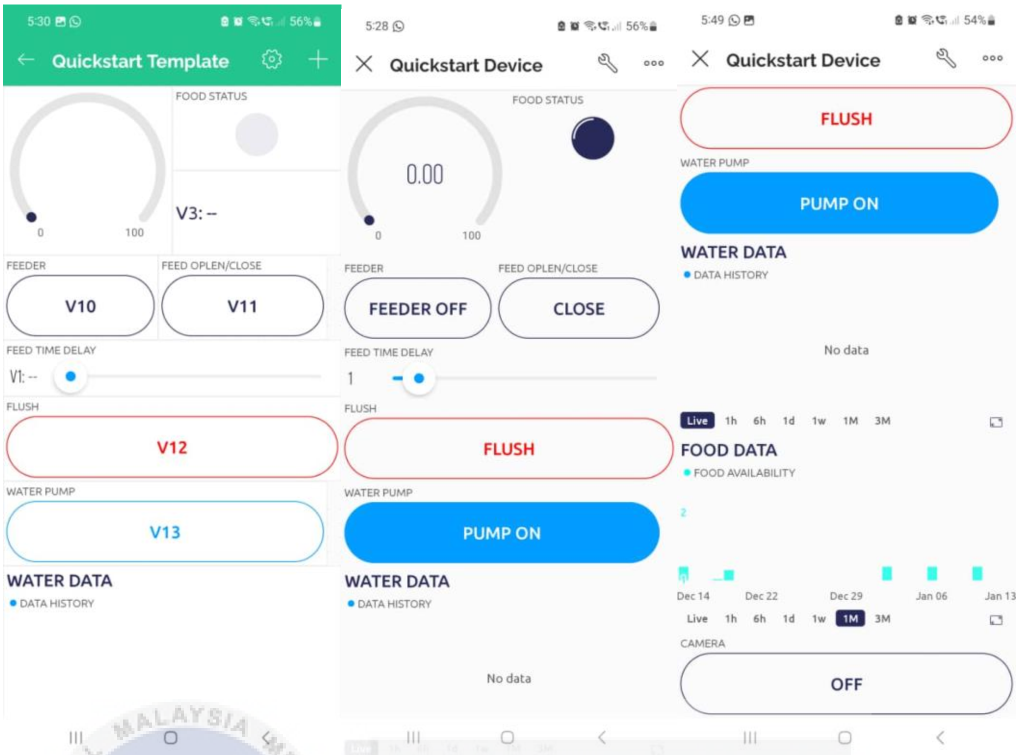


Figure 4.12 User Interface

After sign up an acount on Blynk apps, user need to configure the interface for the apps design. There are few indicator and button have been place in the user interface as shown in Figure 4.12. Each button represent each process that have in this project. The “Feed” button will work for feeding system. The “Pump” button will work for water dispenser system. Meanwhile, the “Flush” button will work for litter box and lastly the “Camera” button for monitoring system.

4.4 Project Result

4.4.1 Feeding System Result

During project testing, a little amount of data was acquired that shows this project can be used to achieve its objectives. For feeding systems, approximately 111g of food is needed to fill the food bowl. Two distinct brands of food, each with a different diameter, were used in the test. To get a reliable result, the test has been done three times. Table 4.3 and Table 4.4 shows the outcome. Data indicates that the amount of food that falls into the bowl depends on the diameter of the pet's food. Figures 4.13 and 4.14 depict how the ProDiet pet food dispenses greater portions amount of pet food compared to Whiskas. It is because ProDiet has smaller diameter.

Table 4.3 Feeding System Data

Food Brand : Whiskas (Diameter = 1.1cm)				
Time (s)	Weight (g)			Average (g)
	1st	2nd	3rd	
1	18	20	17	18.33
2	23	23	22	22.67
3	29	27	28	28.33
4	36	34	37	35.67
5	39	37	39	38.33
6	52	52	50	51.33
7	59	59	57	58.33
8	63	62	63	62.67
9	65	66	67	66.00
10	73	71	70	71.33

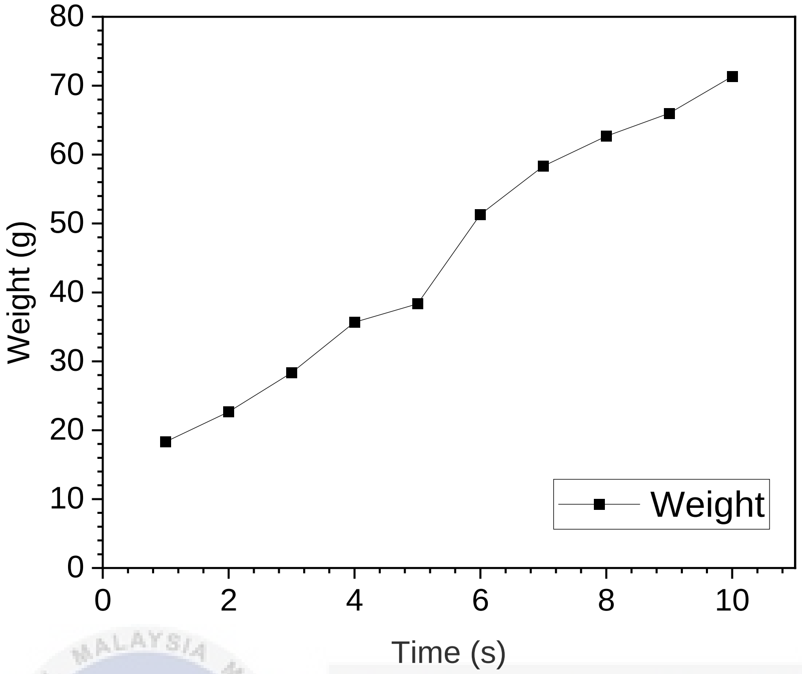


Figure 4.13 Graph of Whiskas Data

Table 4.4 Feeding System Data

Food Brand : ProDiet (Diameter = 0.6cm)				
Time (s)	Weight (g)			Average (g)
	1st	2nd	3rd	
1	16	15	16	15.67
2	21	21	20	20.67
3	27	29	27	27.67
4	37	37	36	36.67
5	41	41	42	41.33
6	47	45	46	46.00
7	60	61	59	60.00
8	70	70	72	70.67
9	77	76	74	75.67
10	83	85	83	83.67

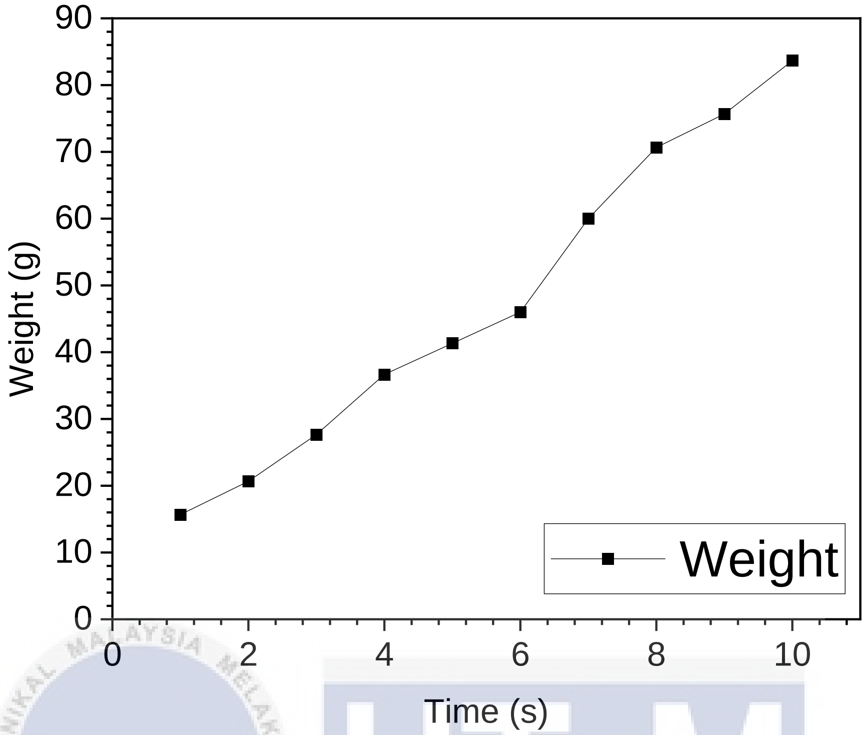


Figure 4.14 Graph of ProDiet Data

4.4.2 Water Dispenser System Result

Table 4.5 Water Flow Data

Time(s)	Volume (ml)			Average (ml)
	1st	2nd	3rd	
1	60	50	65	58.33
2	80	75	80	78.33
3	100	90	95	95
4	130	125	125	126.67
5	160	150	155	155
6	180	170	170	173.33
7	195	200	195	196.67
8	215	220	220	218.33
9	250	250	245	248.33
10	260	265	260	261.67

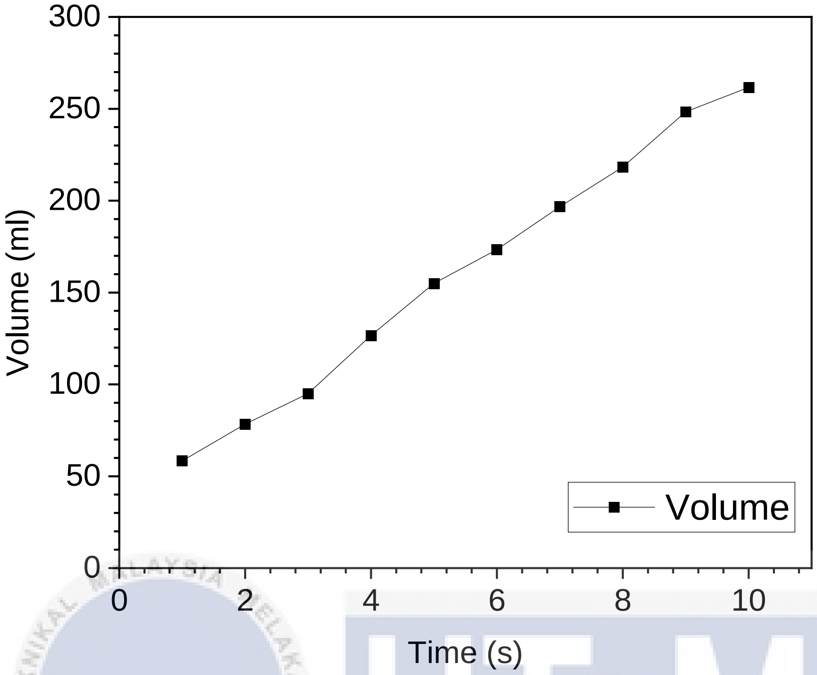


Figure 4.15 Graph of Water Dispenser System

As for water dispenser system, it shows that in Table 4.5 where the volume of water flow into the bowl are getting higher by time. The test was done three times to get a reliable result. It can be conclude that the longer the water pump on the higher volume of water flow into the water bowl. Figure 4.15 shows that volume of water flow are directly proportional to the time of water pump on.

4.4.3 Monitoring System Result

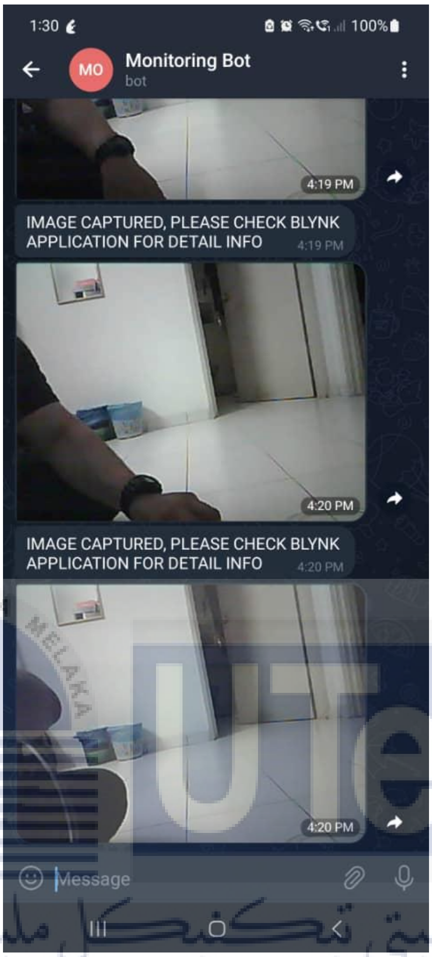


IMAGE CAPTURED, PLEASE CHECK BLYNK APPLICATION FOR DETAIL INFO 4:19 PM

IMAGE CAPTURED, PLEASE CHECK BLYNK APPLICATION FOR DETAIL INFO 4:20 PM

Figure 4.16 Picture Taken By Camera

The system that has been developed contain a monitoring system which is a ESP32-CAMERA that will send picture to the user’s Telegram apps on mobile phone after the user press “Camera” button on the Blynk apps. The result are shown on the Figure 4.16. The camera only works once the user pressed “Camera” button on the Blynk apps. The time taken for the picture to be received in user’s Telegram depends on the internet connection. The stronger the internet connection that ESP32-CAMERA connected, the faster the photo will be received by user.

4.5 Summary

The operation this device works is when the device gets an electrical voltage of 12V. The power window motor will be supplied with 12V from the adapter. Then this device will do scanning or search Wi-Fi to get Internet connection. Once connected to the Wi-Fi and Blynk Application then this device will start to do action, the DC motor rotated as the user press “Flush” on the Blynk apps.

The water pump and servo motor, both actuators will work when the user press feed or pump button on the Blynk apps. It will send the signal to the NodeMCU Esp8266 and the actuator will do their works. The LDR sensor will shows the amount of food inside the container and the water level sensor will measure the volume of water inside the water tank.

Moreover, the microcontroller will also tell the actuator to work accordingly. The ESP32 will need to always be connected to an internet connection for the Blynk application to run. The connection and limitations for each component was analysed, to pursue a successful project and avoid failure in project working.

A successful implementation of components on circuit were done, in hardware setup section as followed by the sketch of circuit from the previous process. The project functions as expected and resulted to problem statement was accomplished with this achievement. A push notification also works well as pictures were able to send to user smartphone via their Telegram once water or food being in low quantity. This successful project of “IoT Based Smart Pet Cage” shown that the result section goes well too.

CHAPTER 5

CONCLUSION

5.1 Introduction

The completion of the project will be discussed in this chapter. As stated earlier in this chapter, the project for PSM 2 is complete, but it must be continued until it is completed. This chapter finishes with a suggestion for future improvements to the project.

5.2 Conclusion

This project aims to develop a reliable automatic pet water and food dispensing system, and pet litter box controlling system where it can help people who have pets to take care of their basic need which as shown in the first objective have been achieved. Study has been done in order to get the information to continue to complete the project. This project has proposed few components to used and it was based on the literature review that have been done. The circuit have been done and simulate. At the end, the circuit works perfectly, and it can be use in the project development. All the parts, including the microcontroller, actuators, and sensors, have been linked and tested to determine if the goals can be met. The second goal of this project, which involves using it to monitor and operate the available actuator via Wi-Fi, is made possible by the development of the mobile apps, Blynk. By obtaining the results of each system that can be used to address the issues outlined in the problem statement, the project's dependability has been demonstrated. The third objective is to evaluate the reliability and performance of the designed smart pet cage system also have been achieved by data analysis have been done. The system itself have been tested through few condition and result have been gathered.

5.3 Project Limitation

The project's design is unreliable in some areas, particularly the water dispenser and litter box systems. The coupling between the litter box and power window motor is not strong enough to withstand the force generated by the sand inside the litter box, which prevents the litter box system from functioning properly. The power window motor is then overly powerful and makes stopping difficult. When it came to stopping, it reacted strongly.

Next, the shape of the water dispenser tank makes it difficult for the water to cease flowing. To stop the flow of water once the water pump shuts off, a check valve must be installed. Lastly, the size of the funnel for the food container is not big enough to make the food drop smoothly.

5.4 Future Works

Despite the fact that this project was a success and solved the problem described in the first chapter, it has several weaknesses that could be rectified. For better future work or development, the project can be improved by considering the following factors.

i) Use suitable and powerful motor for litter box system

ii) Develop a better and smaller design for all system that can be fit into smaller cage.

iii) Upgrade a better light detecting sensor to detect the amount of food inside the container.

iv) Add an extra system for online connection in order to provide better connection.

5.5 Project Potential of Commercialization

After considering every aspect of the project, it is possible to determine that this project can be marketed to organizations that assist animals, such as animal shelters or veterinary clinics. Since the project's primary goal is to assist pet owners in monitoring their animals, it can also be sold to customers that need assistance caring for or monitoring their pets.

REFERENCES

[1] A. A. A. Luayon, G. F. Z. Tolentino, V. K. B. Almazan, P. E. S. Pascual, and M. J. C. Samonte, "PetCare: A smart pet care IoT mobile application," *ACM Int. Conf. Proceeding Ser.*, pp. 427–431, 2019, doi: 10.1145/3306500.3306570.

[2] K. S. Ng, P. Y. Chen, and P. H. Ting, "Smart cage implementation with dependable safety agent for dogs," *Proc. IEEE Pacific Rim Int. Symp. Dependable Comput. PRDC*, pp. 231–232, 2017, doi: 10.1109/PRDC.2017.45.

[3] A. S. A. Putri, A. Rasyid, and A. W. Purwandi, "Smart Cat Home Dengan Sistem Kontrol Yang Menggunakan Aplikasi Telegram," *Jartel*, vol. 8, no. 1, pp. 168-, 2019.

[4] P. P. Cruz, "MiAU: aid system for dogs care," no. June, 2019.

[5] M. A. A. Alias, N. A. Mohd Taib, and N. N. Adnan, "Smart Bran Dispenser for Animal Husbandry Purpose," *Malaysian J. Sci. Heal. Technol.*, vol. 7, no. 1, pp. 56–61, 2021, doi: 10.33102/mjosht.v7i1.159.

[6] T. A. Rao and K. S. Srikanth, "Arduino based Pet feeding automation," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 11, pp. 3443–3445, 2019, doi: 10.35940/ijitee.K2555.0981119.

[7] S. Danial *et al.*, "Politeknik sultan salahuddin abdul aziz shah labelling bottle machine," 2020.

[8] N. Izzatul, N. Binti, M. Razif, N. Binti, and M. Ralim, "Automatic Cat Feeder And Location Tracker," *J. Comput. Technol. Creat. Content*, vol. 5, no. 1, pp. 27–32, 2020, [Online]. Available: http://www.jtec.org.my/index.php/JTeC/article/view/324

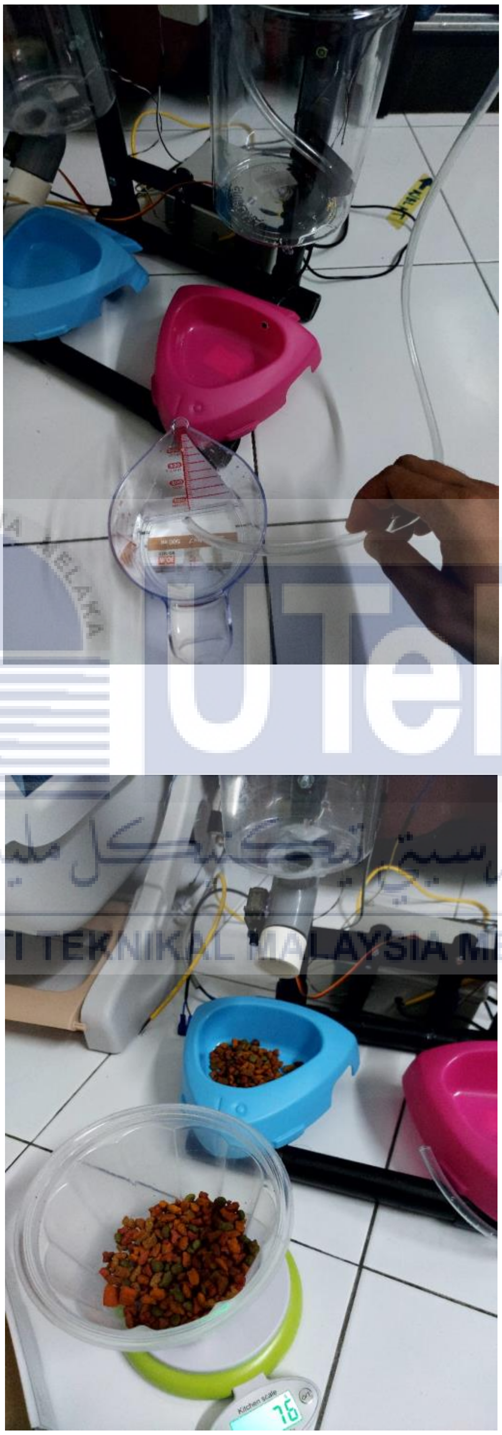
[9] N. Syuhada, B. Muhamad, M. I. A. Syafiqa, A. Binti, M. Firdaus, and B. I. N. Raihan, "Politeknik sultan salahuddin abdul aziz shah automatic cat litter box

(aclb)," 2022.

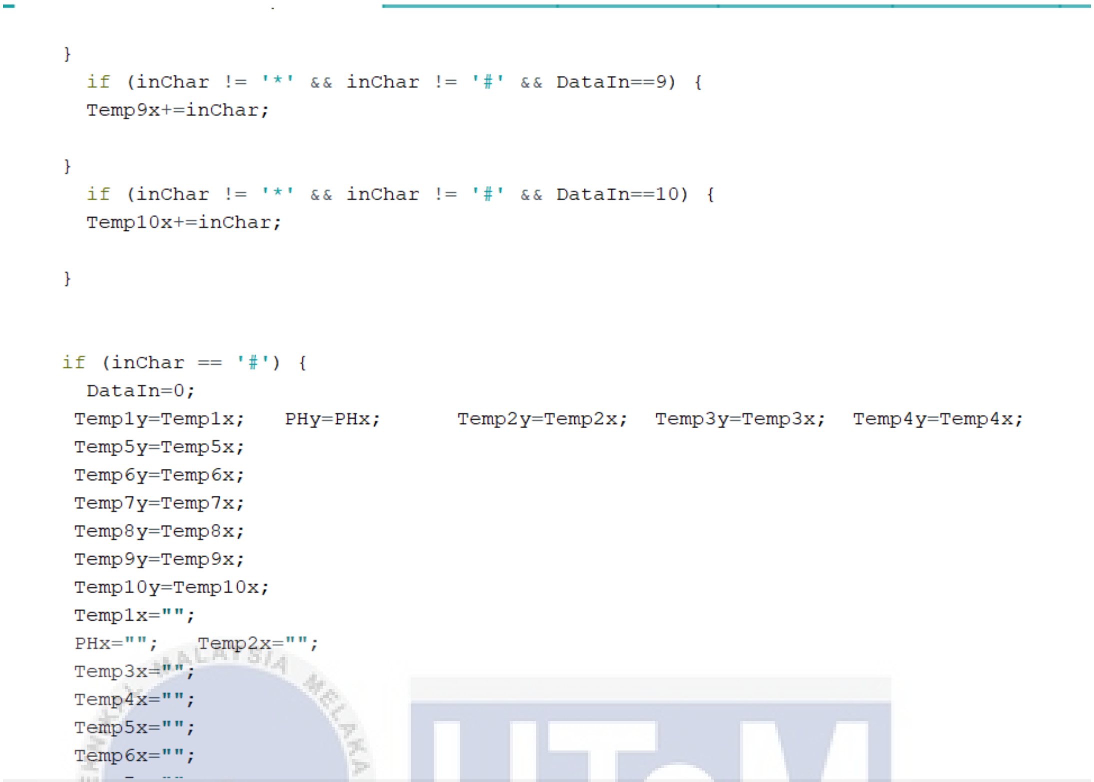
[10] S. Nuraini, B. Sazali, V. Nu, S. Nazierah, A. Binti, and M. Number, *SHAH SMART PET ' S LITTER BOX DEPARTMENT OF MECHANICAL ENGINEERING SHAH SMART PET ' S LITTER BOX DEPARTMENT OF MECHANICAL ENGINEERING*, no. 08. 2022.

APPENDICES

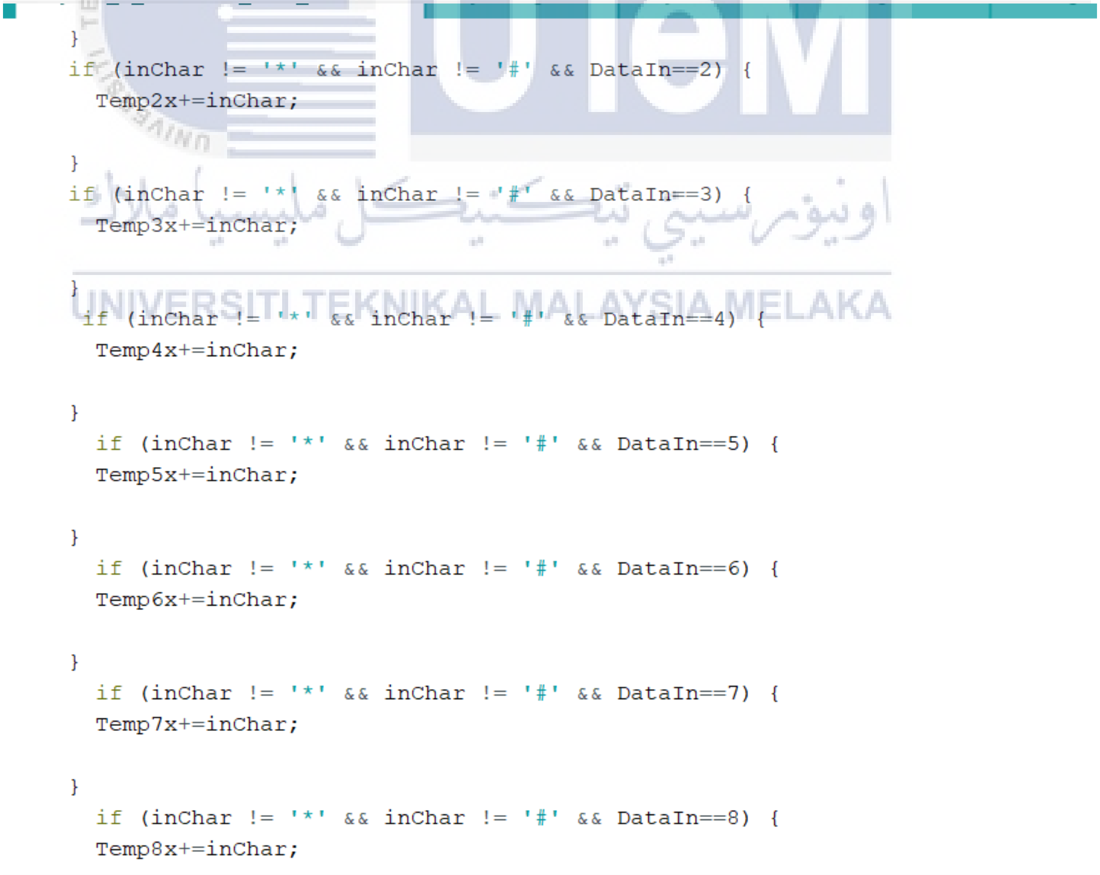
Appendix 1 Process for Testing Pet Feeding System



Appendix 2 Arduino IDE Coding

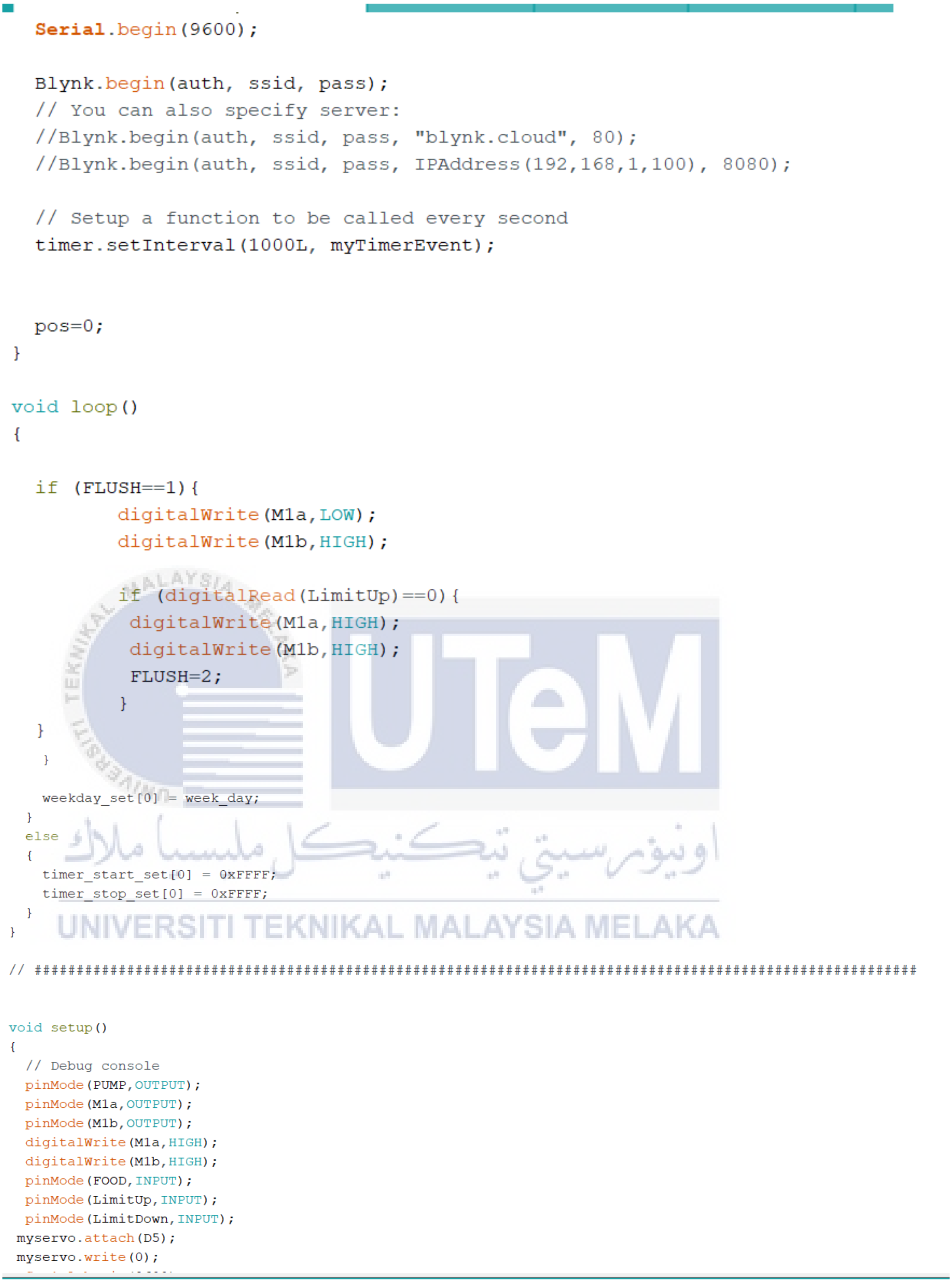


```
}
  if (inChar != '*' && inChar != '#' && DataIn==9) {
  Temp9x+=inChar;

}
  if (inChar != '*' && inChar != '#' && DataIn==10) {
  Temp10x+=inChar;

}


if (inChar == '#') {
  DataIn=0;
 Temp1y=Temp1x;    PHy=PHx;      Temp2y=Temp2x;   Temp3y=Temp3x;   Temp4y=Temp4x;
 Temp5y=Temp5x;
 Temp6y=Temp6x;
 Temp7y=Temp7x;
 Temp8y=Temp8x;
 Temp9y=Temp9x;
 Temp10y=Temp10x;
 Temp1x="";
 PHx="";    Temp2x="";
 Temp3x="";
 Temp4x="";
 Temp5x="";
 Temp6x="";
```



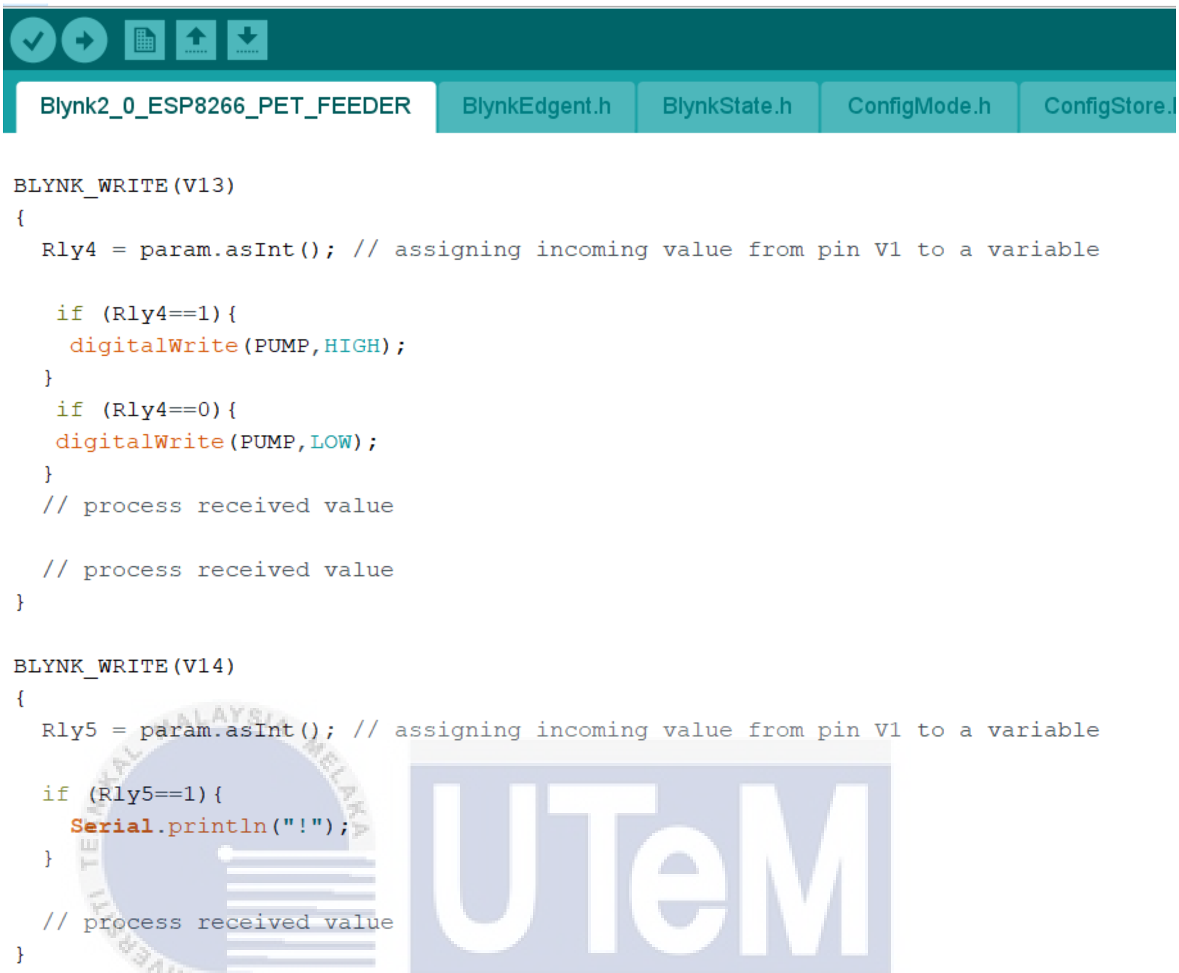
```
}
if (inChar != '*' && inChar != '#' && DataIn==2) {
  Temp2x+=inChar;

}
if (inChar != '*' && inChar != '#' && DataIn==3) {
  Temp3x+=inChar;

}
 if (inChar != '*' && inChar != '#' && DataIn==4) {
  Temp4x+=inChar;

}
  if (inChar != '*' && inChar != '#' && DataIn==5) {
  Temp5x+=inChar;

}
  if (inChar != '*' && inChar != '#' && DataIn==6) {
  Temp6x+=inChar;

}
  if (inChar != '*' && inChar != '#' && DataIn==7) {
  Temp7x+=inChar;

}
  if (inChar != '*' && inChar != '#' && DataIn==8) {
  Temp8x+=inChar;
```

```
    char inChar1 = (char)Serial.read();
  if (inChar1 == '*') {
      DataIn++;

    }

     if (inChar1 == 'Y') {



    }

if (inChar1 == 'X'){

}

    while (DataIn > 0){
        while (Serial.available()) {
    // get the new byte:
    char inChar = (char)Serial.read();
    if (inChar == '*') {
      DataIn++;

    }
    if (inChar != '*' && inChar != '#' && DataIn==1) {
      Temp1x+=inChar;

    if (FLUSH==3){
         digitalWrite(M1a,HIGH);
         digitalWrite(M1b,LOW);

         if (digitalRead(LimitDown)==0){
          digitalWrite(M1a,HIGH);
          digitalWrite(M1b,HIGH);
          FLUSH=4;

         }
  }
      if (FLUSH==5){
        digitalWrite(M1a,LOW);
         digitalWrite(M1b,HIGH);

      delay(700);
          digitalWrite(M1a,HIGH);
          digitalWrite(M1b,HIGH);
          FLUSH=0;
          Blynk.virtualWrite(V12, "0");

  }
  Blynk.run();
  timer.run();
  //---------------------------------------------------
   while (Serial.available()) {
    // get the new byte:
```

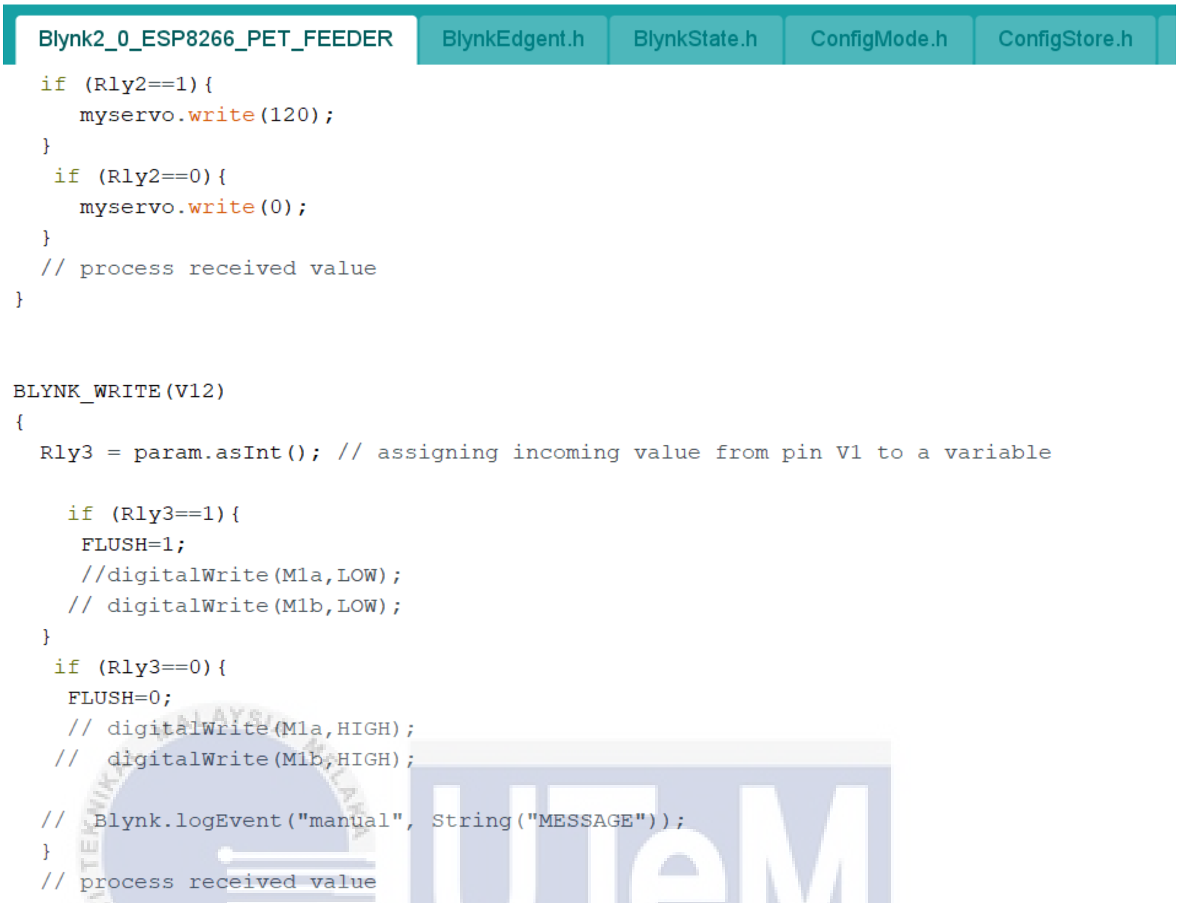


```
  Serial.begin(9600);

  Blynk.begin(auth, ssid, pass);
  // You can also specify server:
  //Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);

  // Setup a function to be called every second
  timer.setInterval(1000L, myTimerEvent);


  pos=0;
}

void loop()
{

  if (FLUSH==1){
         digitalWrite(M1a,LOW);
         digitalWrite(M1b,HIGH);

         if (digitalRead(LimitUp)==0){
          digitalWrite(M1a,HIGH);
          digitalWrite(M1b,HIGH);
          FLUSH=2;
         }
  }
   }

   weekday_set[0] = week_day;
  }
  else
  {
    timer_start_set[0] = 0xFFFF;
    timer_stop_set[0] = 0xFFFF;
  }
}

// ###########################################################################################

void setup()
{
  // Debug console
  pinMode(PUMP,OUTPUT);
  pinMode(M1a,OUTPUT);
  pinMode(M1b,OUTPUT);
  digitalWrite(M1a,HIGH);
  digitalWrite(M1b,HIGH);
  pinMode(FOOD,INPUT);
  pinMode(LimitUp,INPUT);
  pinMode(LimitDown,INPUT);
 myservo.attach(D5);
 myservo.write(0);
```

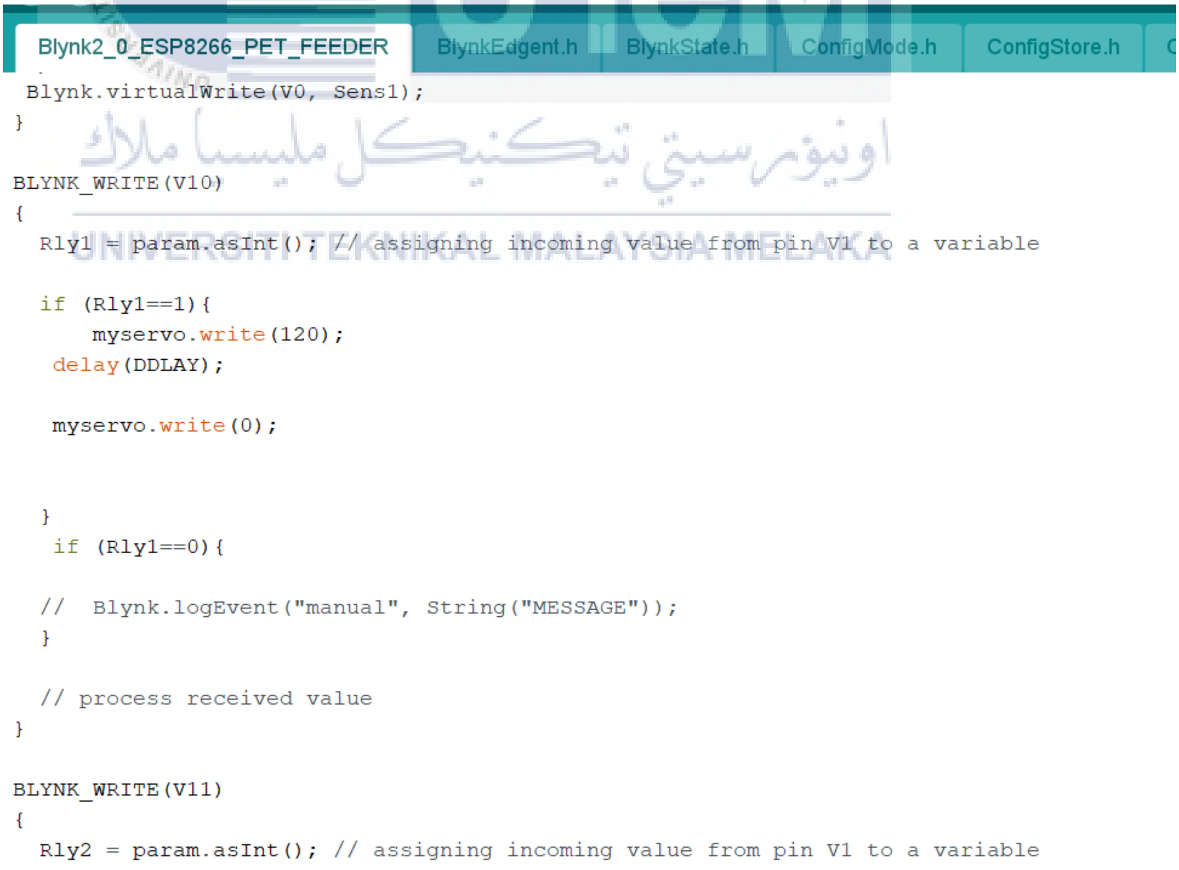
```
  if (t.hasStartTime() && t.hasStopTime() )
  {
    timer_start_set[0] = (t.getStartHour() * 60 * 60) + (t.getStartMinute() * 60) + t.getStartSecond();
    timer_stop_set[0] = (t.getStopHour() * 60 * 60) + (t.getStopMinute() * 60) + t.getStopSecond();

    Serial.println(String("Start Time: ") +
                   t.getStartHour() + ":" +
                   t.getStartMinute() + ":" +
                   t.getStartSecond());

    Serial.println(String("Stop Time: ") +
                   t.getStopHour() + ":" +
                   t.getStopMinute() + ":" +
                   t.getStopSecond());

    for (int i = 1; i <= 7; i++)
    {
      if (t.isWeekdaySelected(i))
      {
        week_day |= (0x01 << (i-1));
        Serial.println(String("Day ") + i + " is selected");
      }
      else
      {
        week_day &= (~(0x01 << (i-1)));
      }
BLYNK_WRITE(V6)
{
  Rly6 = param.asInt(); // assigning incoming value from pin V1 to a variable

  if (Rly6==1){

  }

  // process received value
}


BLYNK_WRITE(V1)
{
 Capasity = param.asInt(); // assigning incoming value from pin V1 to a variable
 //Serial.println(Capasity);


  // process received value
}

BLYNK_WRITE(V9)
{
  unsigned char week_day;

  TimeInputParam t(param);
```



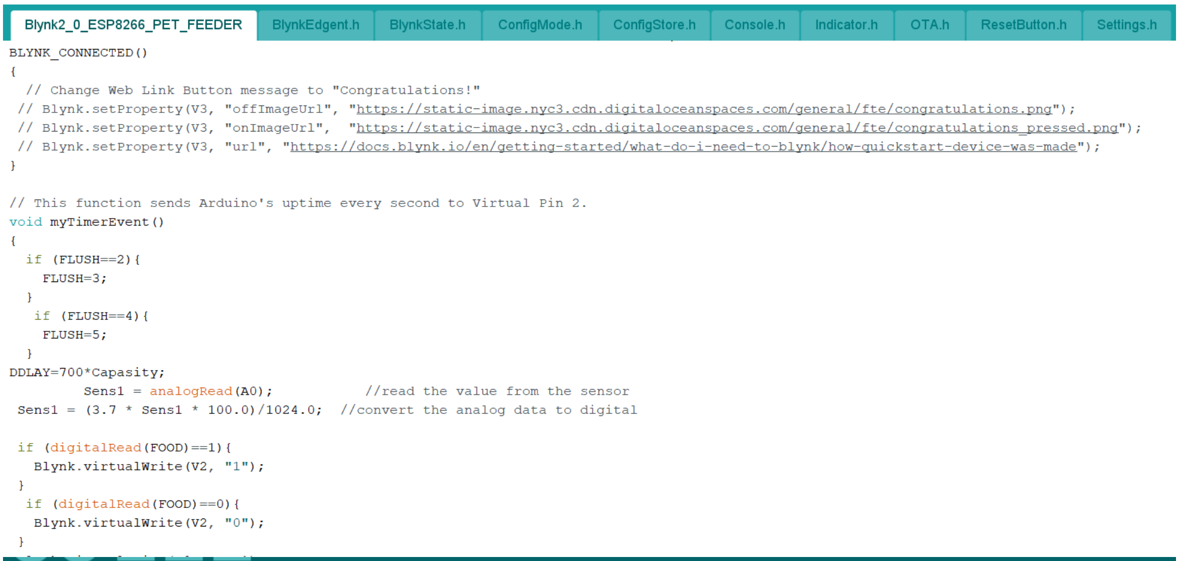
```
BLYNK_WRITE(V13)
{
  Rly4 = param.asInt(); // assigning incoming value from pin V1 to a variable

   if (Rly4==1){
    digitalWrite(PUMP,HIGH);
  }
   if (Rly4==0){
   digitalWrite(PUMP,LOW);
  }
  // process received value

  // process received value
}

BLYNK_WRITE(V14)
{
  Rly5 = param.asInt(); // assigning incoming value from pin V1 to a variable

  if (Rly5==1){
    Serial.println("!");
  }

  // process received value
}
```



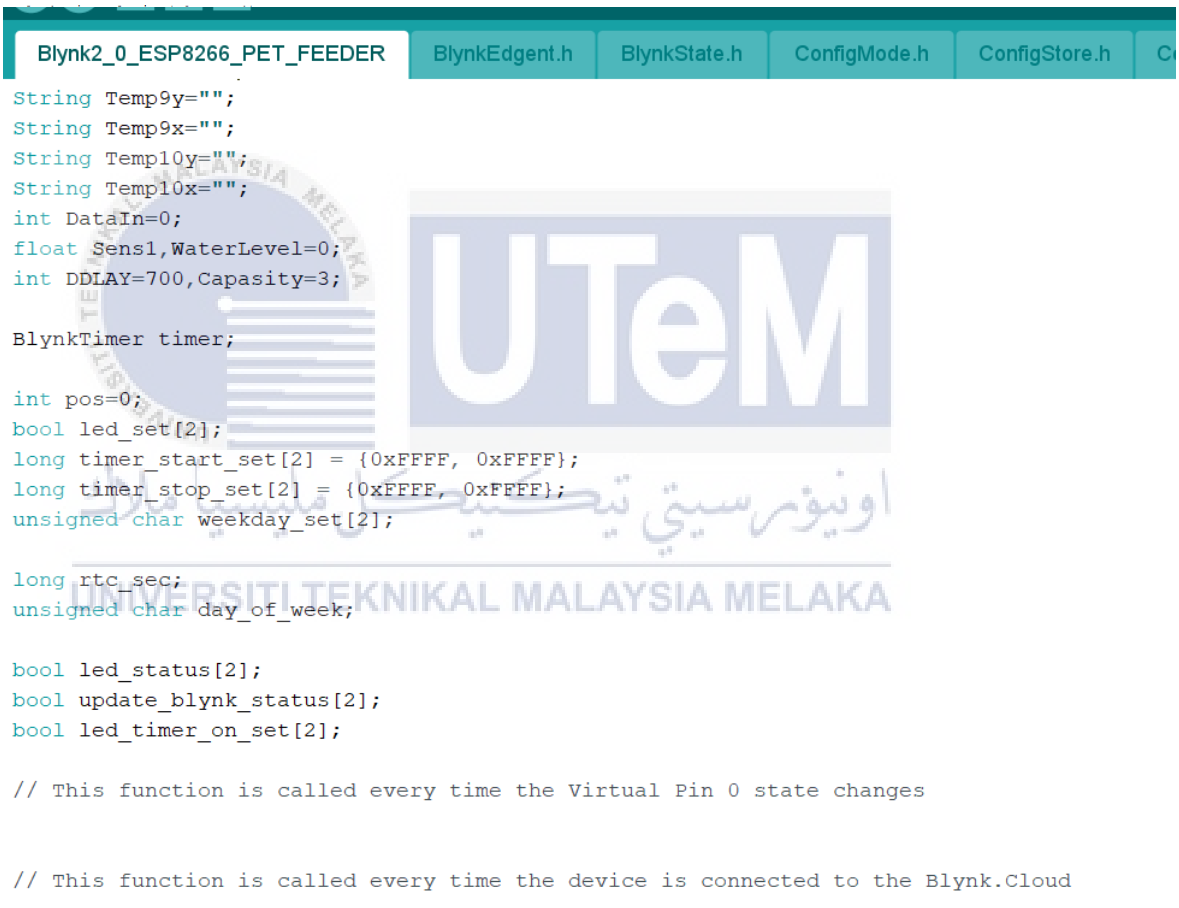
```
  if (Rly2==1){
     myservo.write(120);
  }
   if (Rly2==0){
     myservo.write(0);
  }
  // process received value
}


BLYNK_WRITE(V12)
{
  Rly3 = param.asInt(); // assigning incoming value from pin V1 to a variable

    if (Rly3==1){
     FLUSH=1;
     //digitalWrite(M1a,LOW);
    // digitalWrite(M1b,LOW);
  }
   if (Rly3==0){
    FLUSH=0;
    // digitalWrite(M1a,HIGH);
   //  digitalWrite(M1b,HIGH);

  //  Blynk.logEvent("manual", String("MESSAGE"));
  }
  // process received value
```



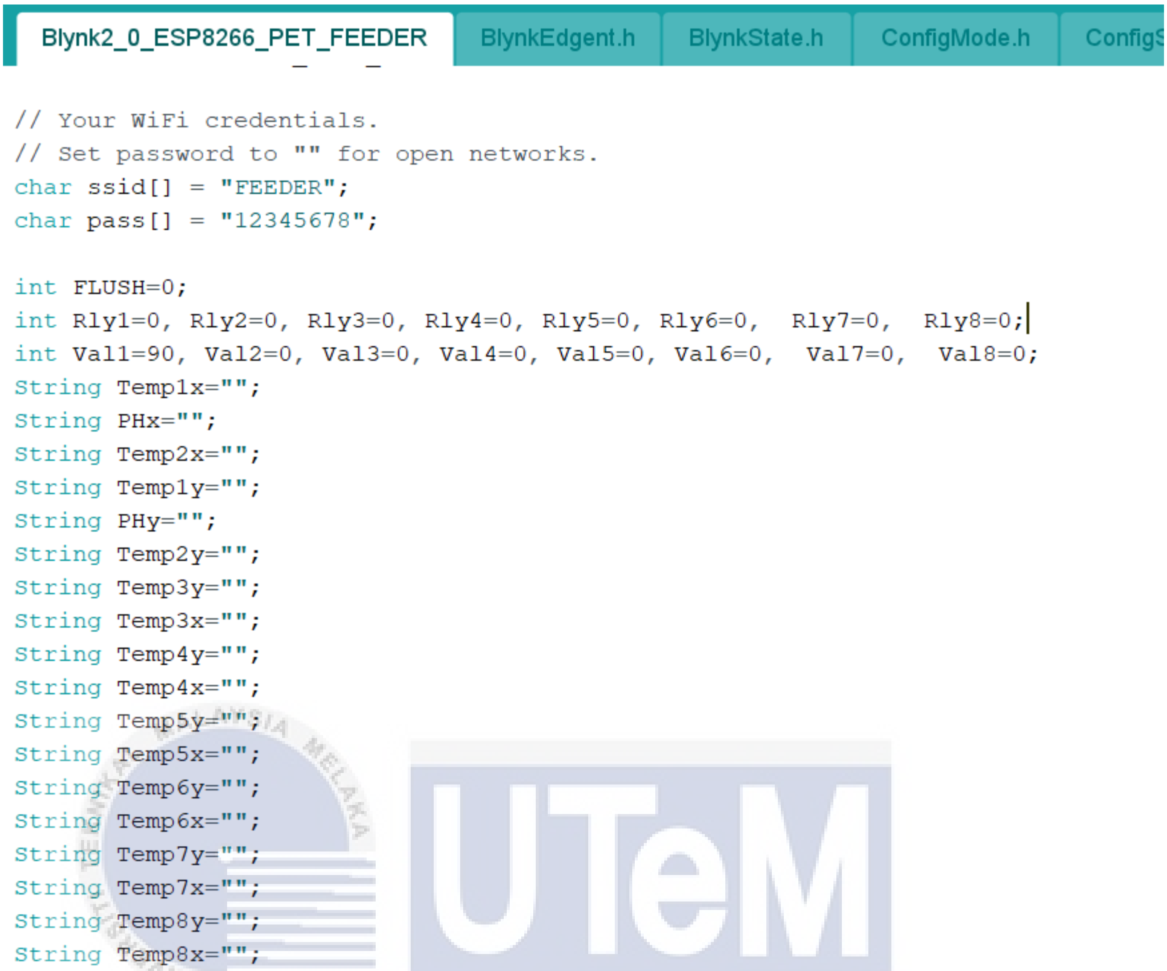
```
 Blynk.virtualWrite(V0, Sens1);
}

BLYNK_WRITE(V10)
{
  Rly1 = param.asInt(); // assigning incoming value from pin V1 to a variable

  if (Rly1==1){
      myservo.write(120);
   delay(DDLAY);

   myservo.write(0);


  }
   if (Rly1==0){

  //  Blynk.logEvent("manual", String("MESSAGE"));
  }

  // process received value
}

BLYNK_WRITE(V11)
{
  Rly2 = param.asInt(); // assigning incoming value from pin V1 to a variable
```



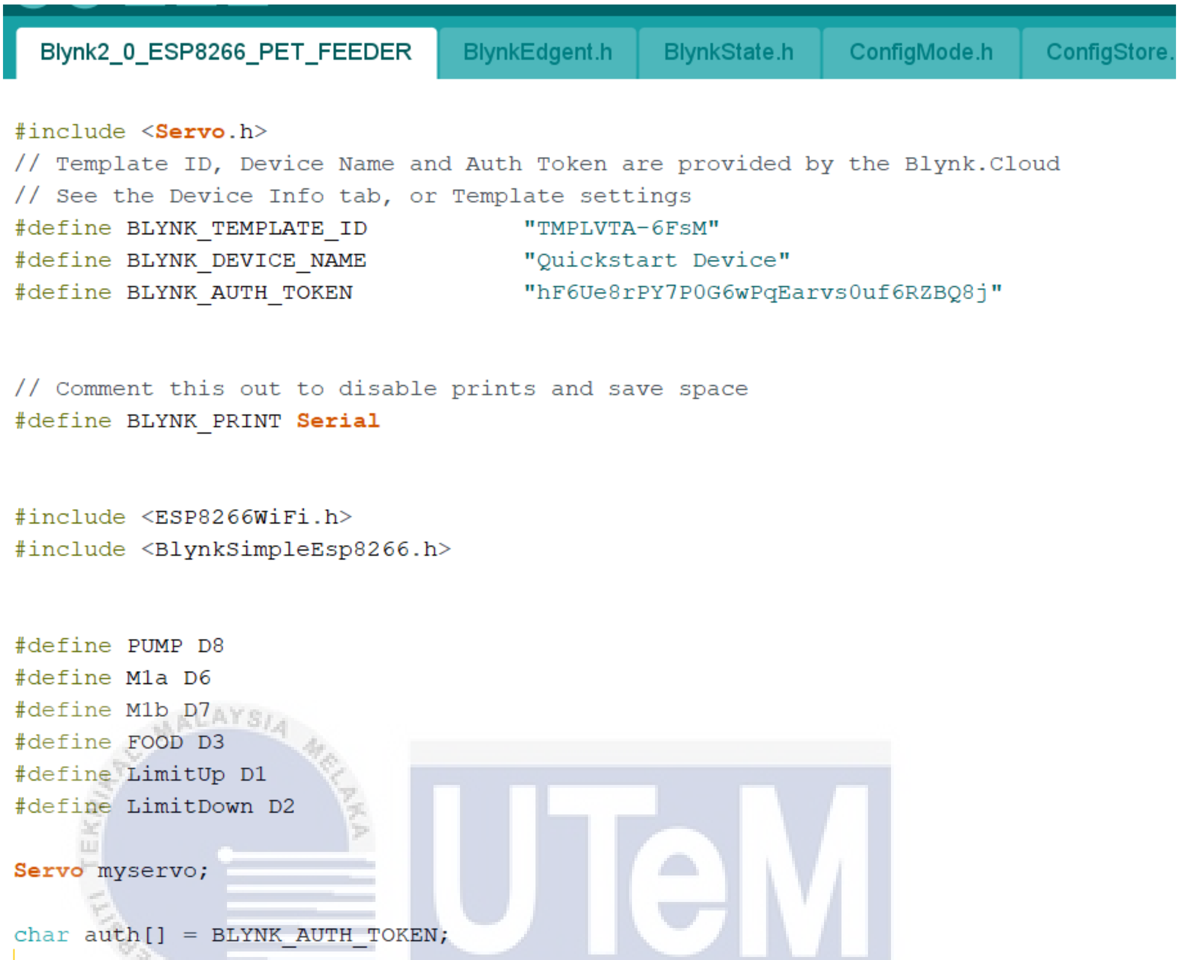
```
BLYNK_CONNECTED()
{
  // Change Web Link Button message to "Congratulations!"
 // Blynk.setProperty(V3, "offImageUrl", "https://static-image.nyc3.cdn.digitaloceanspaces.com/general/fte/congratulations.png");
 // Blynk.setProperty(V3, "onImageUrl",  "https://static-image.nyc3.cdn.digitaloceanspaces.com/general/fte/congratulations_pressed.png");
 // Blynk.setProperty(V3, "url", "https://docs.blynk.io/en/getting-started/what-do-i-need-to-blynk/how-quickstart-device-was-made");
}

// This function sends Arduino's uptime every second to Virtual Pin 2.
void myTimerEvent()
{
  if (FLUSH==2){
    FLUSH=3;
  }
   if (FLUSH==4){
    FLUSH=5;
  }
DDLAY=700*Capasity;
         Sens1 = analogRead(A0);           //read the value from the sensor
 Sens1 = (3.7 * Sens1 * 100.0)/1024.0;  //convert the analog data to digital

 if (digitalRead(FOOD)==1){
   Blynk.virtualWrite(V2, "1");
 }
  if (digitalRead(FOOD)==0){
   Blynk.virtualWrite(V2, "0");
 }
```



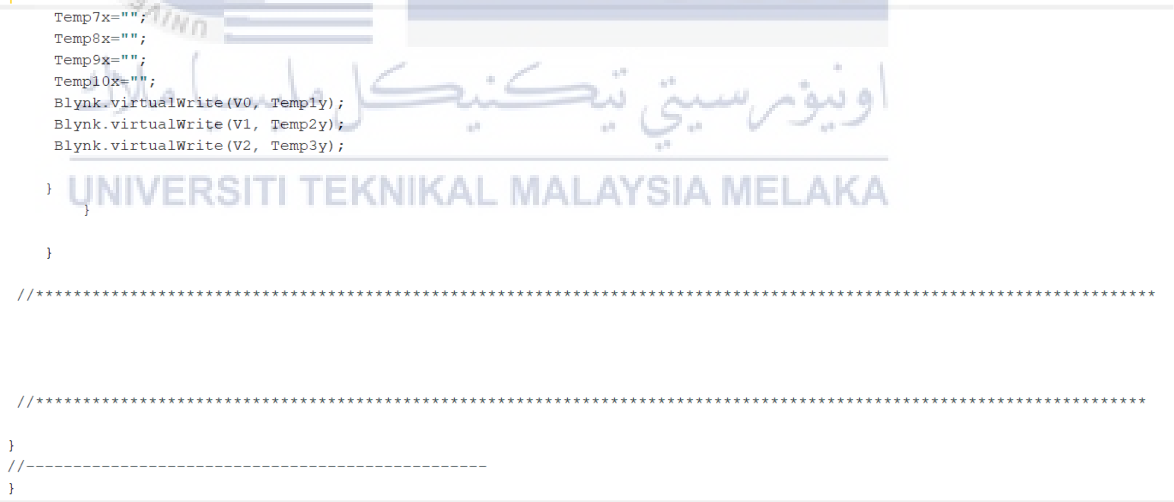
```
String Temp9y="";
String Temp9x="";
String Temp10y="";
String Temp10x="";
int DataIn=0;
float Sens1,WaterLevel=0;
int DDLAY=700,Capasity=3;

BlynkTimer timer;

int pos=0;
bool led_set[2];
long timer_start_set[2] = {0xFFFF, 0xFFFF};
long timer_stop_set[2] = {0xFFFF, 0xFFFF};
unsigned char weekday_set[2];

long rtc_sec;
unsigned char day_of_week;

bool led_status[2];
bool update_blynk_status[2];
bool led_timer_on_set[2];

// This function is called every time the Virtual Pin 0 state changes


// This function is called every time the device is connected to the Blynk.Cloud
```



```
// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "FEEDER";
char pass[] = "12345678";

int FLUSH=0;
int Rly1=0, Rly2=0, Rly3=0, Rly4=0, Rly5=0, Rly6=0,  Rly7=0,  Rly8=0;
int Val1=90, Val2=0, Val3=0, Val4=0, Val5=0, Val6=0,  Val7=0,  Val8=0;
String Temp1x="";
String PHx="";
String Temp2x="";
String Temp1y="";
String PHy="";
String Temp2y="";
String Temp3y="";
String Temp3x="";
String Temp4y="";
String Temp4x="";
String Temp5y="";
String Temp5x="";
String Temp6y="";
String Temp6x="";
String Temp7y="";
String Temp7x="";
String Temp8y="";
String Temp8x="";
```

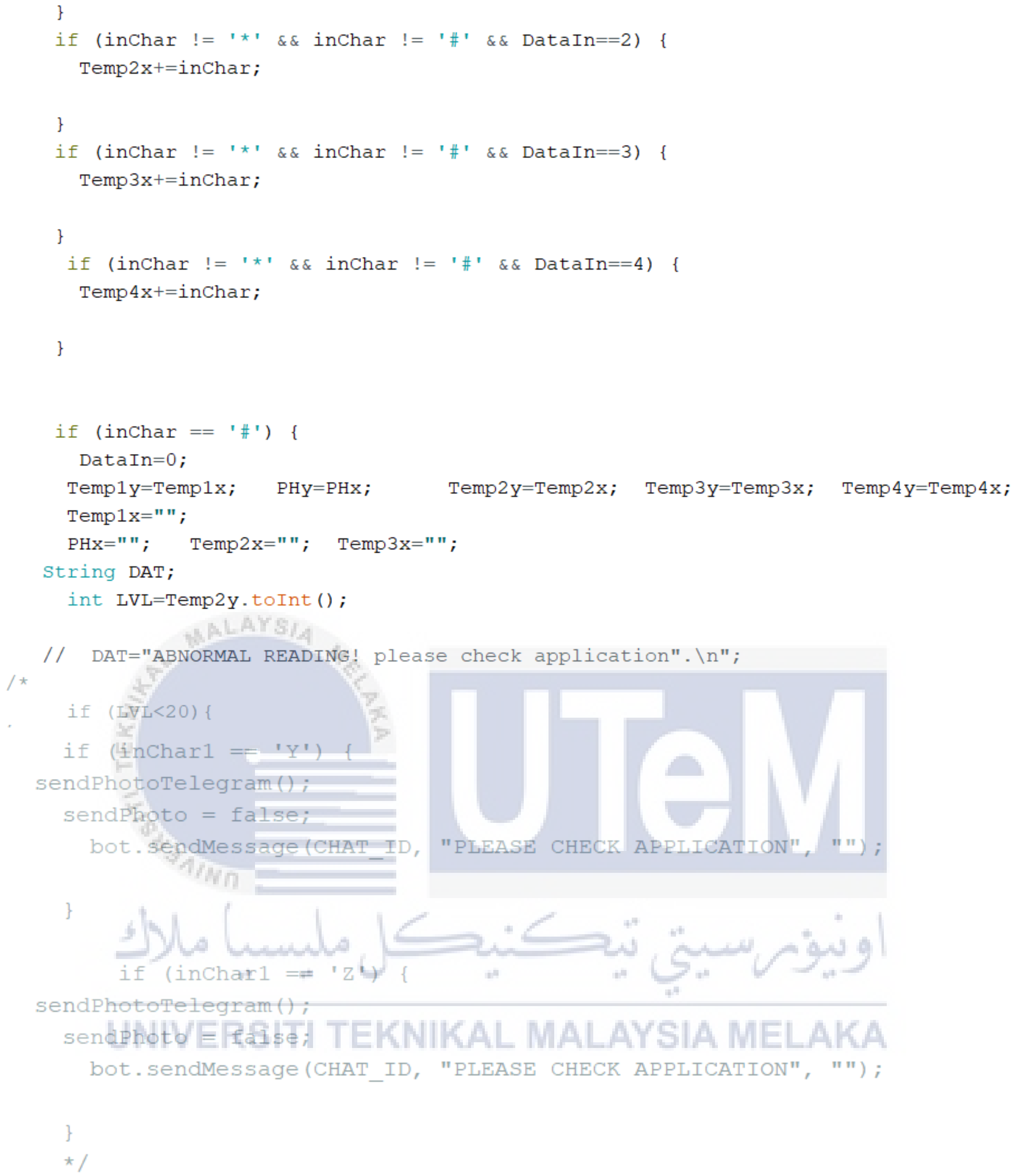


```
#include <Servo.h>
// Template ID, Device Name and Auth Token are provided by the Blynk.Cloud
// See the Device Info tab, or Template settings
#define BLYNK_TEMPLATE_ID           "TMPLVTA-6FsM"
#define BLYNK_DEVICE_NAME           "Quickstart Device"
#define BLYNK_AUTH_TOKEN            "hF6Ue8rPY7P0G6wPqEarvs0uf6RZBQ8j"


// Comment this out to disable prints and save space
#define BLYNK_PRINT Serial


#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>


#define PUMP D8
#define M1a D6
#define M1b D7
#define FOOD D3
#define LimitUp D1
#define LimitDown D2

Servo myservo;

char auth[] = BLYNK_AUTH_TOKEN;
```

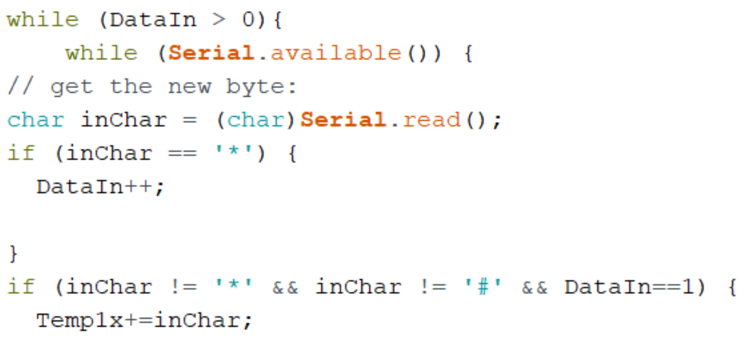


```
    Temp7x="";
    Temp8x="";
    Temp9x="";
    Temp10x="";
    Blynk.virtualWrite(V0, Temp1y);
    Blynk.virtualWrite(V1, Temp2y);
    Blynk.virtualWrite(V2, Temp3y);

   }
      }

   }
 //*******************************************************************************************************


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

}
//--------------------------------------------------
}
```

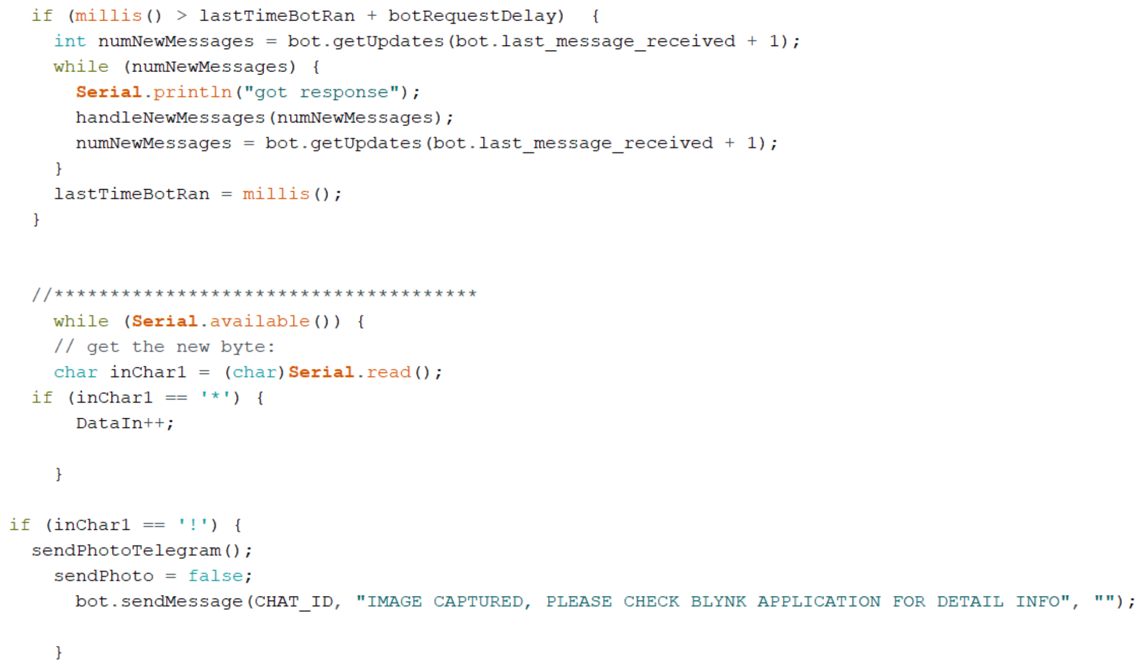
1. ESP32-CAMERA



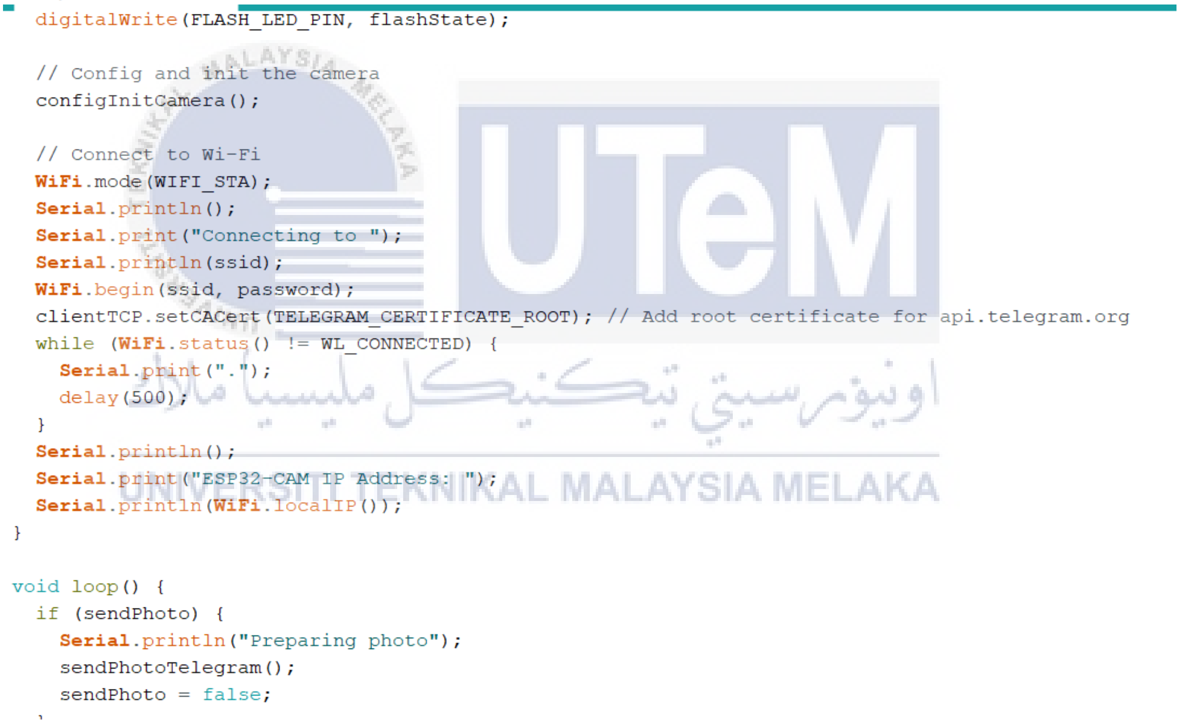
```
  }
  if (inChar != '*' && inChar != '#' && DataIn==2) {
    Temp2x+=inChar;

  }
  if (inChar != '*' && inChar != '#' && DataIn==3) {
    Temp3x+=inChar;

  }
   if (inChar != '*' && inChar != '#' && DataIn==4) {
    Temp4x+=inChar;

  }


  if (inChar == '#') {
    DataIn=0;
   Temp1y=Temp1x;    PHy=PHx;      Temp2y=Temp2x;  Temp3y=Temp3x;  Temp4y=Temp4x;
   Temp1x="";
   PHx="";    Temp2x="";  Temp3x="";
 String DAT;
   int LVL=Temp2y.toInt();

 //  DAT="ABNORMAL READING! please check application".\n";
/*
   if (LVL<20){
   if (inChar1 == 'Y') {
sendPhotoTelegram();
  sendPhoto = false;
    bot.sendMessage(CHAT_ID, "PLEASE CHECK APPLICATION", "");

   }
       if (inChar1 == 'Z') {
sendPhotoTelegram();
  sendPhoto = false;
    bot.sendMessage(CHAT_ID, "PLEASE CHECK APPLICATION", "");

   }
   */
```



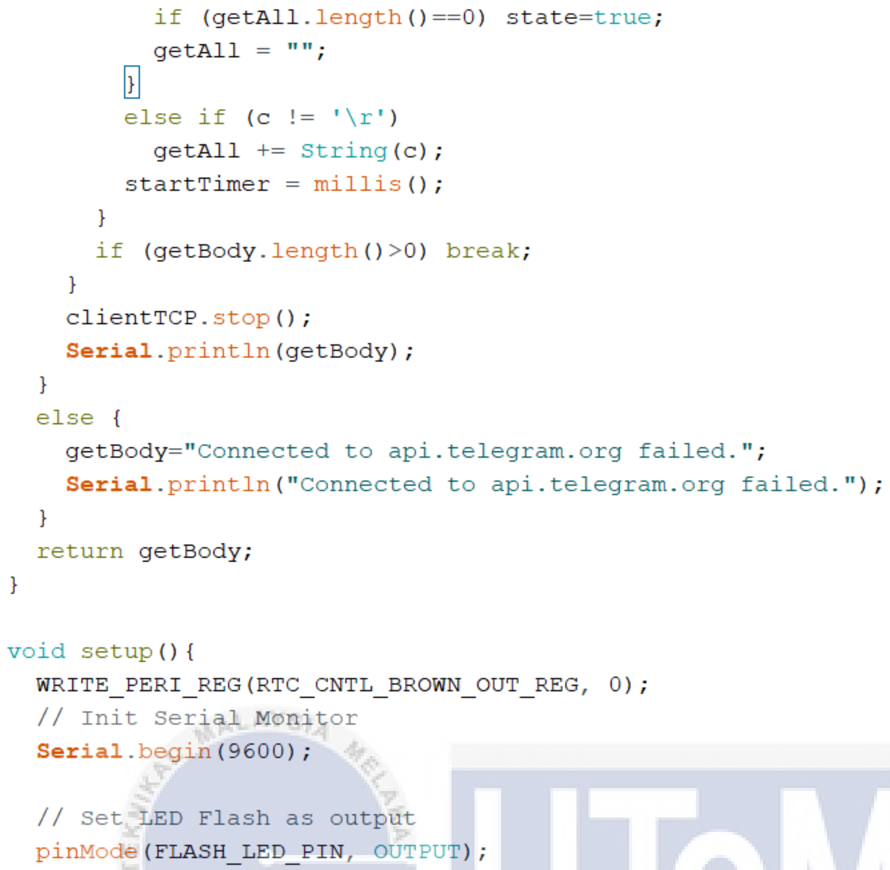
```
  while (DataIn > 0){
      while (Serial.available()) {
  // get the new byte:
  char inChar = (char)Serial.read();
  if (inChar == '*') {
    DataIn++;

  }
  if (inChar != '*' && inChar != '#' && DataIn==1) {
    Temp1x+=inChar;
```



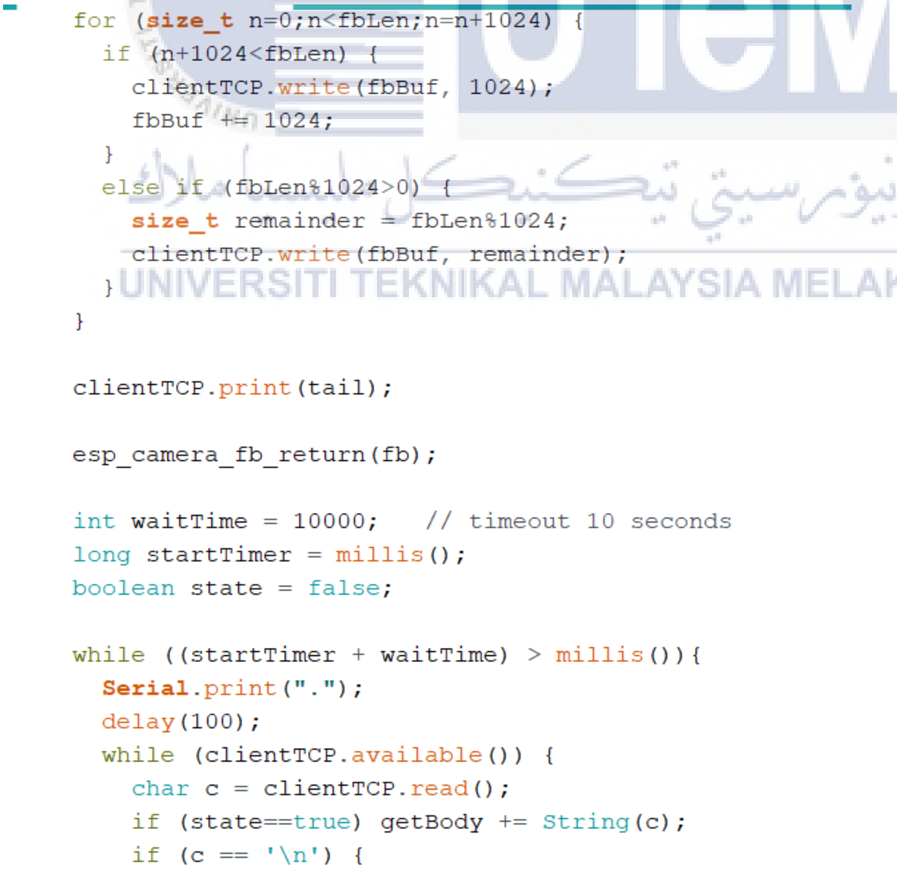
```
  if (millis() > lastTimeBotRan + botRequestDelay)  {
    int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
    while (numNewMessages) {
      Serial.println("got response");
      handleNewMessages(numNewMessages);
      numNewMessages = bot.getUpdates(bot.last_message_received + 1);
    }
    lastTimeBotRan = millis();
  }


  //****************************************
    while (Serial.available()) {
    // get the new byte:
    char inChar1 = (char)Serial.read();
  if (inChar1 == '*') {
      DataIn++;

    }

if (inChar1 == '!') {
  sendPhotoTelegram();
    sendPhoto = false;
      bot.sendMessage(CHAT_ID, "IMAGE CAPTURED, PLEASE CHECK BLYNK APPLICATION FOR DETAIL INFO", "");

    }
```



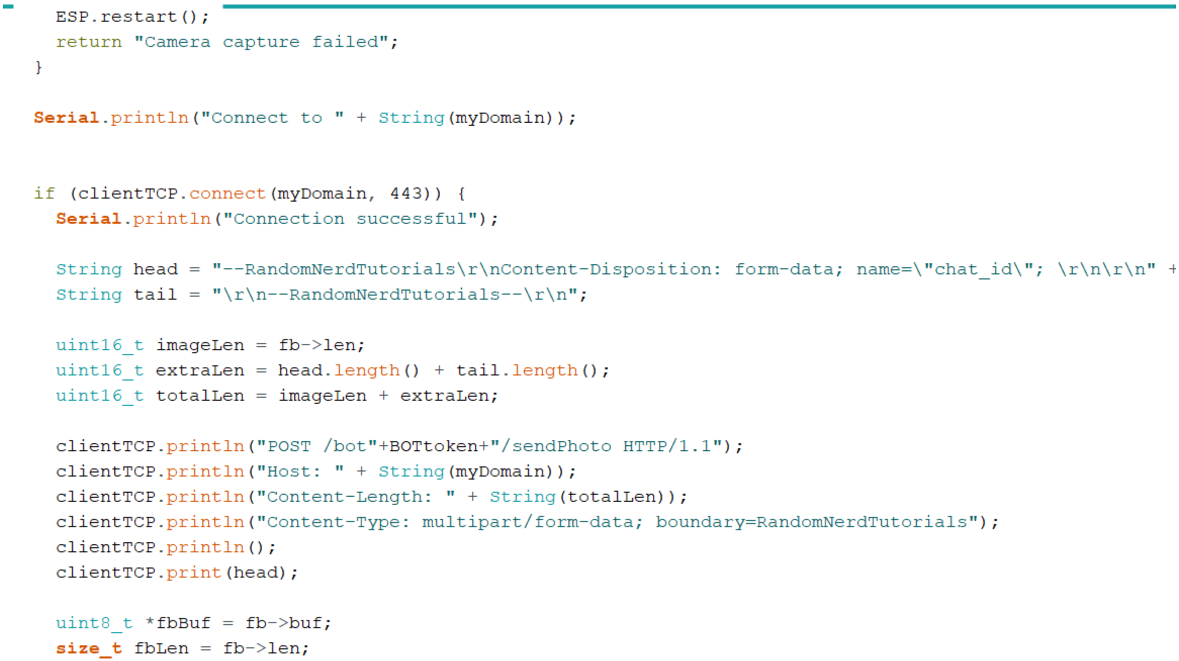
```
  digitalWrite(FLASH_LED_PIN, flashState);

  // Config and init the camera
  configInitCamera();

  // Connect to Wi-Fi
  WiFi.mode(WIFI_STA);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  clientTCP.setCACert(TELEGRAM_CERTIFICATE_ROOT); // Add root certificate for api.telegram.org
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("ESP32-CAM IP Address: ");
  Serial.println(WiFi.localIP());
}

void loop() {
  if (sendPhoto) {
    Serial.println("Preparing photo");
    sendPhotoTelegram();
    sendPhoto = false;
  }
```



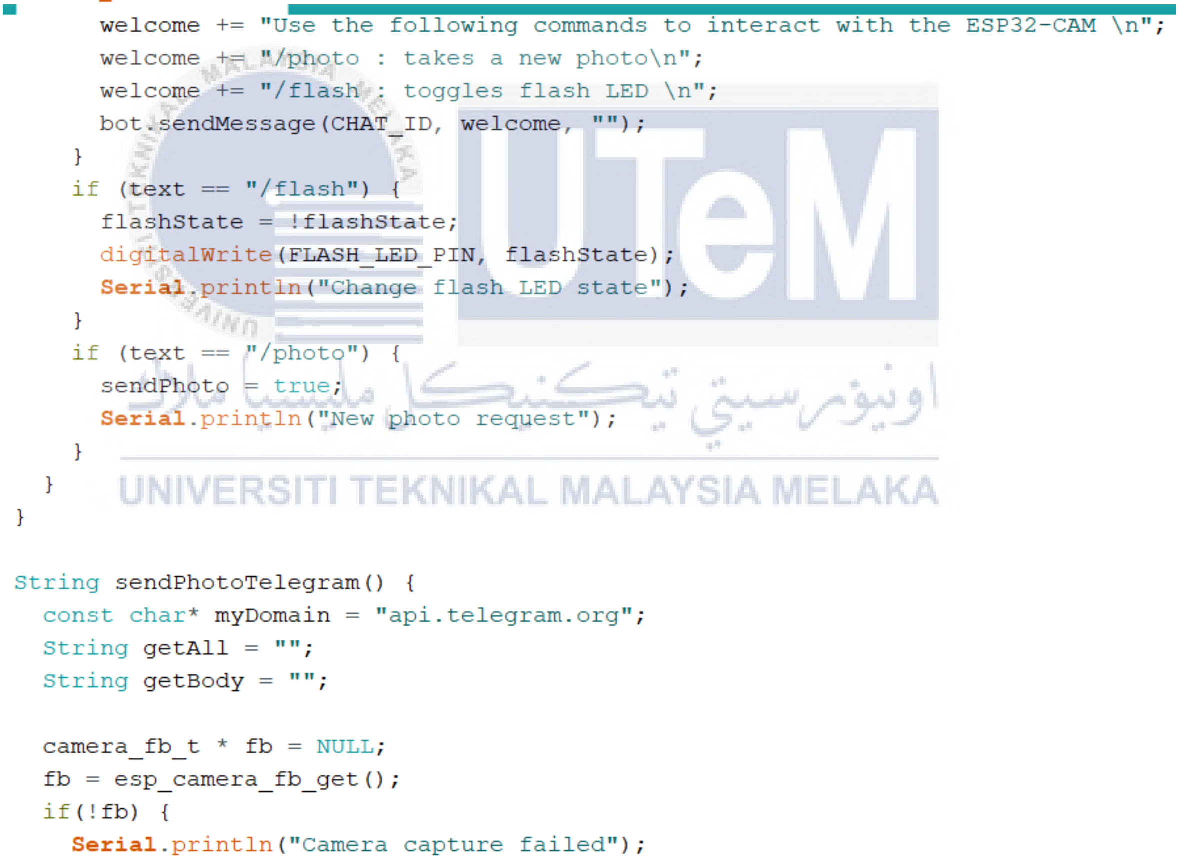
```
          if (getAll.length()==0) state=true;
          getAll = "";
        }
        else if (c != '\r')
          getAll += String(c);
        startTimer = millis();
      }
      if (getBody.length()>0) break;
    }
    clientTCP.stop();
    Serial.println(getBody);
  }
  else {
    getBody="Connected to api.telegram.org failed.";
    Serial.println("Connected to api.telegram.org failed.");
  }
  return getBody;
}

void setup(){
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);
  // Init Serial Monitor
  Serial.begin(9600);

  // Set LED Flash as output
  pinMode(FLASH_LED_PIN, OUTPUT);
```



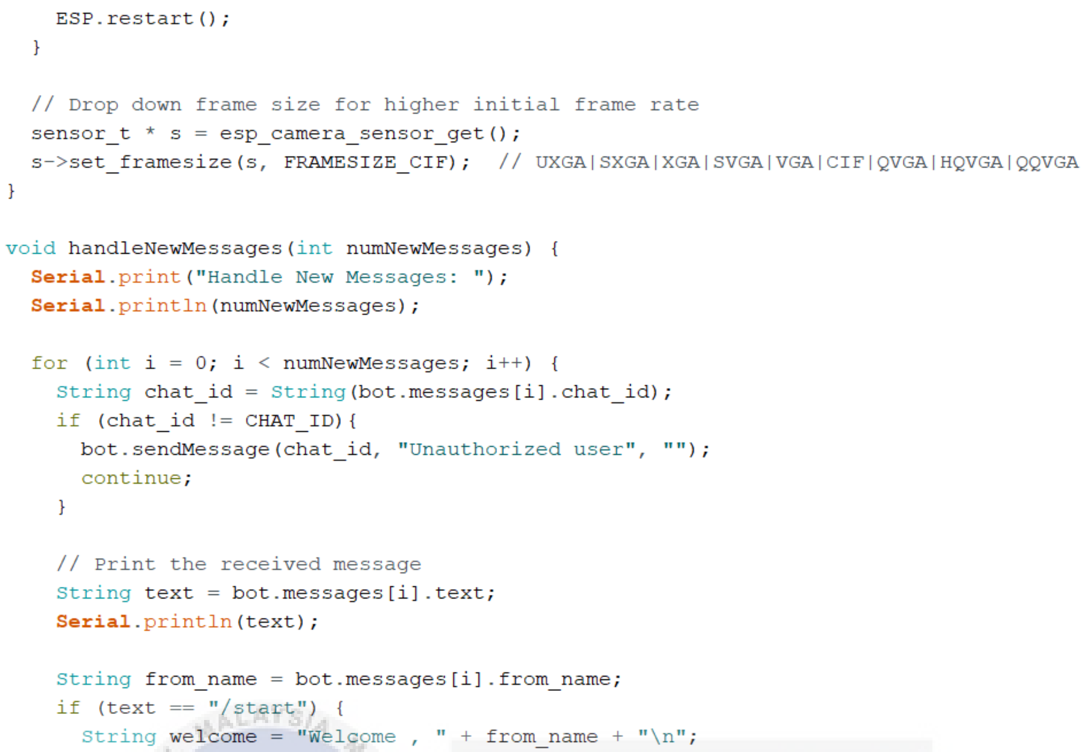
```
    for (size_t n=0;n<fbLen;n=n+1024) {
      if (n+1024<fbLen) {
        clientTCP.write(fbBuf, 1024);
        fbBuf += 1024;
      }
      else if (fbLen%1024>0) {
        size_t remainder = fbLen%1024;
        clientTCP.write(fbBuf, remainder);
      }
    }

    clientTCP.print(tail);

    esp_camera_fb_return(fb);

    int waitTime = 10000;   // timeout 10 seconds
    long startTimer = millis();
    boolean state = false;

    while ((startTimer + waitTime) > millis()){
      Serial.print(".");
      delay(100);
      while (clientTCP.available()) {
        char c = clientTCP.read();
        if (state==true) getBody += String(c);
        if (c == '\n') {
```



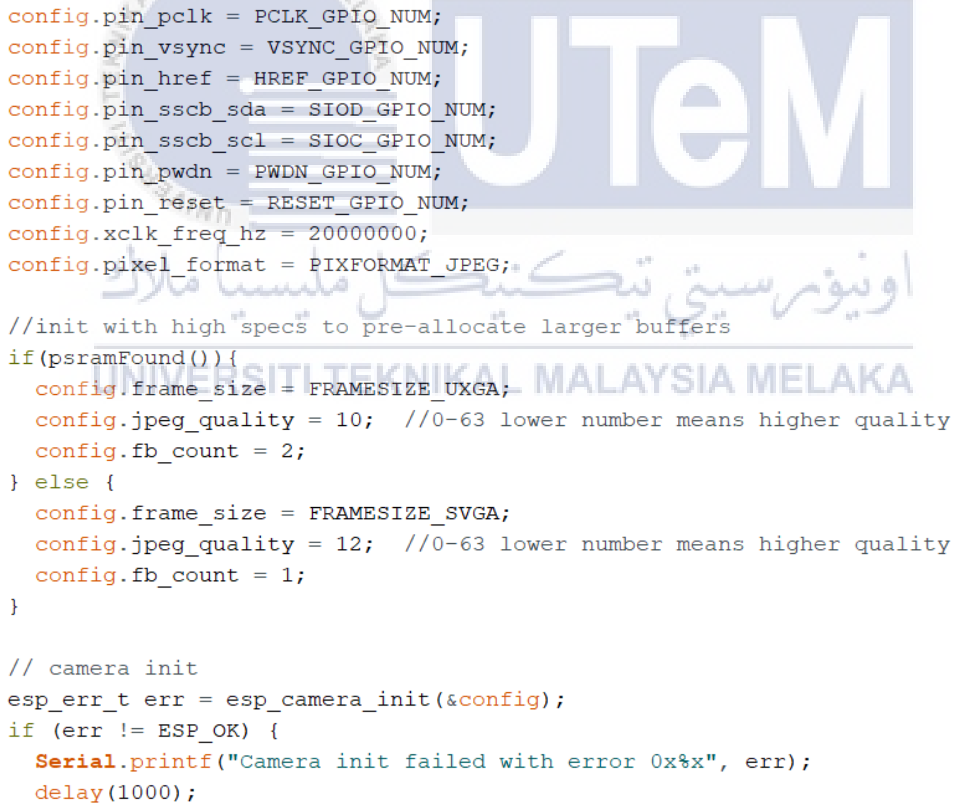
```
    ESP.restart();
    return "Camera capture failed";
  }

  Serial.println("Connect to " + String(myDomain));


  if (clientTCP.connect(myDomain, 443)) {
    Serial.println("Connection successful");

    String head = "--RandomNerdTutorials\r\nContent-Disposition: form-data; name=\"chat_id\"; \r\n\r\n" +
    String tail = "\r\n--RandomNerdTutorials--\r\n";

    uint16_t imageLen = fb->len;
    uint16_t extraLen = head.length() + tail.length();
    uint16_t totalLen = imageLen + extraLen;

    clientTCP.println("POST /bot"+BOTtoken+"/sendPhoto HTTP/1.1");
    clientTCP.println("Host: " + String(myDomain));
    clientTCP.println("Content-Length: " + String(totalLen));
    clientTCP.println("Content-Type: multipart/form-data; boundary=RandomNerdTutorials");
    clientTCP.println();
    clientTCP.print(head);

    uint8_t *fbBuf = fb->buf;
    size_t fbLen = fb->len;
```



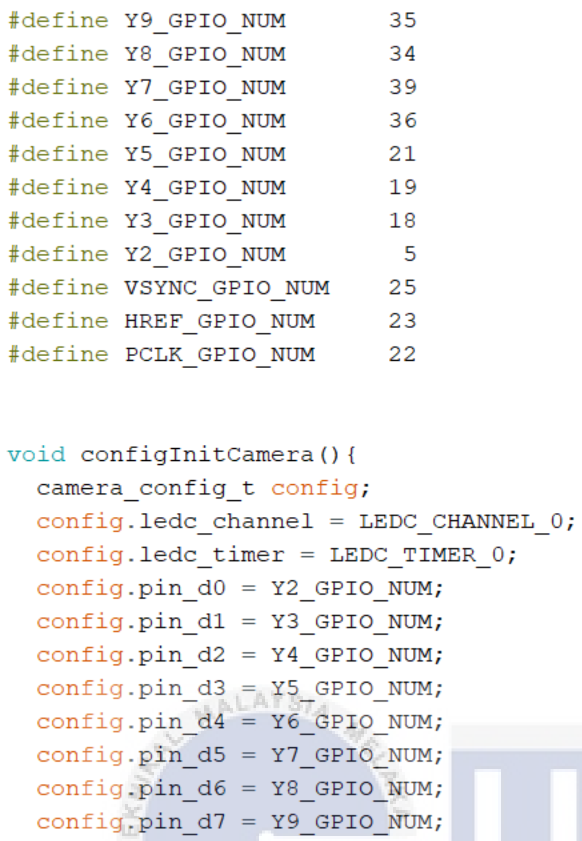
```
      welcome += "Use the following commands to interact with the ESP32-CAM \n";
      welcome += "/photo : takes a new photo\n";
      welcome += "/flash : toggles flash LED \n";
      bot.sendMessage(CHAT_ID, welcome, "");
    }
    if (text == "/flash") {
      flashState = !flashState;
      digitalWrite(FLASH_LED_PIN, flashState);
      Serial.println("Change flash LED state");
    }
    if (text == "/photo") {
      sendPhoto = true;
      Serial.println("New photo request");
    }
  }
}

String sendPhotoTelegram() {
  const char* myDomain = "api.telegram.org";
  String getAll = "";
  String getBody = "";

  camera_fb_t * fb = NULL;
  fb = esp_camera_fb_get();
  if(!fb) {
    Serial.println("Camera capture failed");
```



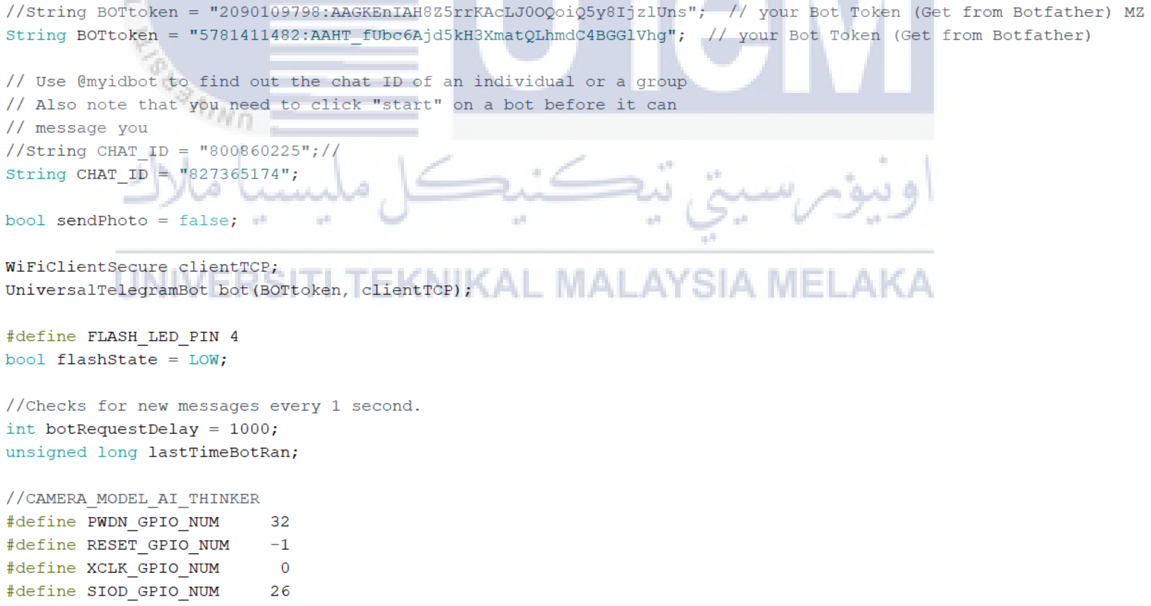
```
    ESP.restart();
  }

  // Drop down frame size for higher initial frame rate
  sensor_t * s = esp_camera_sensor_get();
  s->set_framesize(s, FRAMESIZE_CIF);  // UXGA|SXGA|XGA|SVGA|VGA|CIF|QVGA|HQVGA|QQVGA
}

void handleNewMessages(int numNewMessages) {
  Serial.print("Handle New Messages: ");
  Serial.println(numNewMessages);

  for (int i = 0; i < numNewMessages; i++) {
    String chat_id = String(bot.messages[i].chat_id);
    if (chat_id != CHAT_ID){
      bot.sendMessage(chat_id, "Unauthorized user", "");
      continue;
    }

    // Print the received message
    String text = bot.messages[i].text;
    Serial.println(text);

    String from_name = bot.messages[i].from_name;
    if (text == "/start") {
      String welcome = "Welcome , " + from_name + "\n";
```



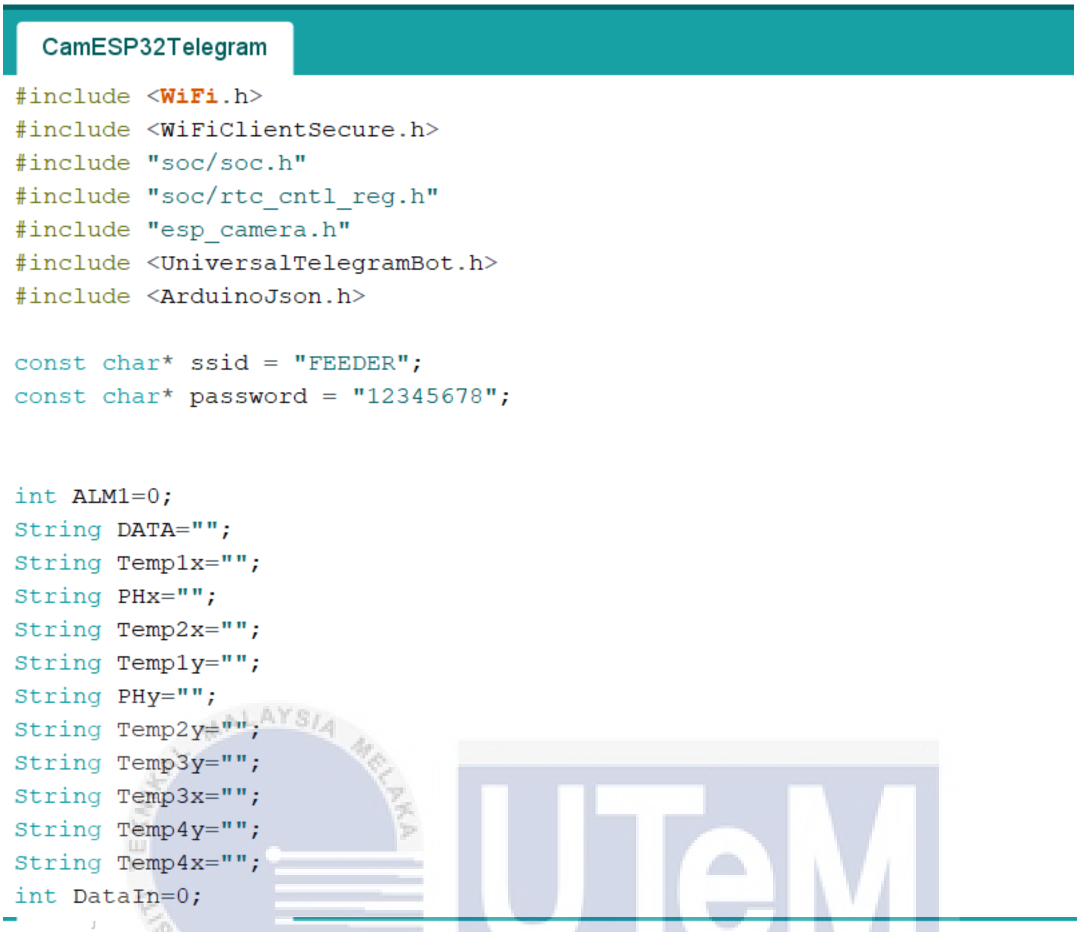
```
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sscb_sda = SIOD_GPIO_NUM;
  config.pin_sscb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
  config.xclk_freq_hz = 20000000;
  config.pixel_format = PIXFORMAT_JPEG;

  //init with high specs to pre-allocate larger buffers
  if(psramFound()){
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;  //0-63 lower number means higher quality
    config.fb_count = 2;
  } else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;  //0-63 lower number means higher quality
    config.fb_count = 1;
  }

  // camera init
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    delay(1000);
```



```
#define Y9_GPIO_NUM       35
#define Y8_GPIO_NUM       34
#define Y7_GPIO_NUM       39
#define Y6_GPIO_NUM       36
#define Y5_GPIO_NUM       21
#define Y4_GPIO_NUM       19
#define Y3_GPIO_NUM       18
#define Y2_GPIO_NUM        5
#define VSYNC_GPIO_NUM    25
#define HREF_GPIO_NUM     23
#define PCLK_GPIO_NUM     22


void configInitCamera(){
  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
```



```
//String BOTtoken = "2090109798:AAGKEnIAH8Z5rrKAcLJ0OQoiQ5y8IjzlUns";  // your Bot Token (Get from Botfather) MZ
String BOTtoken = "5781411482:AAHT_fUbc6Ajd5kH3XmatQLhmdC4BGGlVhg";  // your Bot Token (Get from Botfather)

// Use @myidbot to find out the chat ID of an individual or a group
// Also note that you need to click "start" on a bot before it can
// message you
//String CHAT_ID = "800860225";//
String CHAT_ID = "827365174";

bool sendPhoto = false;

WiFiClientSecure clientTCP;
UniversalTelegramBot bot(BOTtoken, clientTCP);

#define FLASH_LED_PIN 4
bool flashState = LOW;

//Checks for new messages every 1 second.
int botRequestDelay = 1000;
unsigned long lastTimeBotRan;

//CAMERA_MODEL_AI_THINKER
#define PWDN_GPIO_NUM     32
#define RESET_GPIO_NUM    -1
#define XCLK_GPIO_NUM      0
#define SIOD_GPIO_NUM     26
```



```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"
#include "esp_camera.h"
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>

const char* ssid = "FEEDER";
const char* password = "12345678";


int ALM1=0;
String DATA="";
String Temp1x="";
String PHx="";
String Temp2x="";
String Temp1y="";
String PHy="";
String Temp2y="";
String Temp3y="";
String Temp3x="";
String Temp4y="";
String Temp4x="";
int DataIn=0;
```

```
      }
        if (LVL>20){
      DAT=Temp1y + "  LEVEL:" + Temp2y + "  ABNORMAL!!" + ".\n";
       }
       */
       sendPhotoTelegram();
      sendPhoto = false;
  //     bot.sendMessage(CHAT_ID, "WARNING, "");
        bot.sendMessage(CHAT_ID, DAT);




      }
              }

      }

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




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

}
//*****************************************************************************
```

APPENDICES

Final Report PSM 2

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

PAGE 40

PAGE 41

PAGE 42

PAGE 43

PAGE 44

PAGE 45

PAGE 46

PAGE 47

PAGE 48

PAGE 49

PAGE 50

PAGE 51

PAGE 52

PAGE 53

PAGE 54

PAGE 55

PAGE 56

PAGE 57

PAGE 58

PAGE 59

PAGE 60

PAGE 61

PAGE 62

PAGE 63

PAGE 64

PAGE 65

PAGE 66

PAGE 67

PAGE 68

PAGE 69

PAGE 70

PAGE 71

PAGE 72

PAGE 73

PAGE 74

PAGE 75

PAGE 76

PAGE 77

PAGE 78

PAGE 79

PAGE 80

PAGE 81

PAGE 82

PAGE 83

PAGE 84

PAGE 85

PAGE 86

PAGE 87

PAGE 88

PAGE 89

PAGE 90

PAGE 91

PAGE 92

PAGE 93

PAGE 94

PAGE 95

PAGE 96

PAGE 97

PAGE 98