IoT Smart Feeder with Quantity Food Monitoring

Prepared by:
LUTFI HADI BIN MOHD NASRAN

Supervised by:
NORAN ZAHRINE BIN ZAINAL

Kolej Kemahiran Tinggi MARA
Petaling Jaya Selangor Darul Ehsan
AUG-DEC 2022

"I hereby declare that I have read this project report and in my opinion this project report is sufficient in terms of scope and quality for the award of the Diploma in Electronic Engineering (Internet of Things)".

Signature :
Name of advisor :
Date :

I declare that this project report titled “IoT Smart Feeder with Quantity Food Monitoring” is the result of my own work/research except as cited in the references. The project report has not been accepted for any diploma and is not concurrently submitted in candidature of any other diploma.

‘

Signature :
Name of Author :
Date :

ABSTRACT

Machines can automate autonomously, but the Internet of Things (IoT) allows for the addition of monitoring and regulating features [1]. As the world becomes more networked, the IoT builds a vast network of devices that routinely exchange data. In addition to occurring in corporations and organizations on a global scale, this interconnectedness also occurs in private households. Consumers who enjoy having all of their gadgets connected for the purposes of increased convenience, comfort, energy efficiency, and most importantly, personalization—which is one of the project's focus points—are becoming more and more interested in smart home gadgets and devices. With the aid of automation in electronics and IoT, the experience for the user becomes much more personalized. A pet feeding system that is separated into two primary sectors, comprising measurements and a control unit, is suggested in this project report. Periodic assessments of the system's functioning are made based on the food level inside the tank. The system may function in smart mode on the control unit, allowing the user to

specify the food's energy content inside the container. The appropriate daily calorie intake for the chosen cat determines how much food is needed. The flow rate of the feeder is controlled by an IoT smart feeder and quantity food monitoring system depending on motion, cat kind, and cat motion [10]. The flow rate of the big outlet, which can only be used in smart mode, was determined to be 320.11 cm^3, whereas the flow rate of the tiny outputs is 81.62 cm^3.

ABSTRAK

Automasi boleh dilakukan secara bebas oleh mesin, tetapi ia boleh dipertingkatkan dengan ciri pemantauan dan kawalan dengan bantuan IoT[1], apabila dunia menjadi semakin saling berkaitan, Internet Perkara (IoT) mencipta rangkaian peranti yang meluas yang kerap bertukar-tukar data. Walaupun kesalinghubungan ini berlaku dalam perniagaan dan di seluruh organisasi di peringkat global, ia juga berlaku di rumah individu. Peranti dan gajet rumah pintar menjadi lebih popular di kalangan pengguna yang menikmati semua peranti mereka disambungkan untuk memenuhi tujuan peningkatan keyakinan, keselesaan, kecekapan tenaga dan yang paling penting pemperibadian yang merupakan salah satu titik fokus pada projek ini, dengan bantuan automasi elektronik dan IoT pengalaman menjadi lebih diperibadikan untuk pengguna. Penyelidikan ini mencadangkan sistem makanan haiwan peliharaan yang terbahagi kepada dua sektor utama termasuk ukuran dan unit kawalan. Prestasi sistem pengukuran berkala tahap makanan di dalam tangki. Pada unit kawalan, sistem boleh beroperasi di bawah mod pintar di mana

pengguna boleh menentukan fakta tenaga makanan di dalam bekas dan jenis kucing.. Jumlah makanan yang betul adalah berdasarkan tenaga yang mencukupi yang perlu diambil oleh kucing terpilih setiap hari. Pemberian Makanan Pintar dengan Pengawalan kuantiti makanan adalah berdasarkan gerakan, jenis kucing, gerakan kucing untuk mengawal kadar aliran penyuap[10. Alur keluar besar hanya digunakan dalam mod pintar di mana kadar alirannya dikira sebagai 320.11 cm^3s manakala alur keluar kecil membenarkan kadar aliran 81.62 cm^3s.

DEDICATION

Especially for:

My parents, my advisor and my relatives who have always be there to support and provide me some new ideas for my project development, as well as catalyst motivation to ensure that I fulfil the requirements of the project. To my Lecturers, who has always been willing to assist me in times of need and to provide logical explanations for me to understand the process of work that must be completed by Practical.

ACKNOWLEDGEMENTS

I thank Allah, the Beneficent, and the Merciful, for giving me the patience to completing this project.

I am overwhelmed in all humbleness and gratefulness to acknowledge my depth to all those who have helped me to put these ideas, well above the level of simplicity and into something concrete.

I would like to express my special thanks of gratitude to my Project advisor Encik Noran Zahrine Bin Zainal and as well as our project co-coordinator, Encik Hafif Bin Abdul Hamid who gave me the golden opportunity to do this Final year project on the topic IoT Smart Feeder with Food Quality Monitoring, he also helped me in doing a lot of Research and I came to know about so many new things. I am really thankful to them.

I would like to thank my parents who helped me a lot in provide funding, collecting data and guiding me from time to time in making this project, despite of their busy schedules, they gave me different ideas in making this project unique in their own way

TABLE OF CONTENTS

LIST OF TABLES:

LIST OF FIGURES:

LIST OF APPENDICES:

Chapter 1

Introduction

1.1 Problem Statements & Solutions

Smart feeder machine statement consists of several statements: -

i. Lack of time to take care of pet's food when you're busy at your workplace

By using PIR Motion Sensor your pet will have no trouble asking for food

Instead, they will simply walk in the distance of the sensor and food will come out immediately by itself

ii. Concern food will be not enough for one day unaware

By using Ultrasonic Sensor, we can monitor food quantity wirelessly using our own device.

iii. Quality of food will be bad if it kept in the container for a long time

By Using DHT 11, we can monitor humidity of the container so that we can replace the food to a new one if the one we put earlier be bad because of humidity in the contain

1.2 Literature Review

For Smart Feeder Machine there's several literature reviews have been cited:

Title	Author	Link URL	Description
Make a Smart Automatic Pet Feeder with Arduino Uno	DevicePlus Editorial Team	https://www.deviceplus.com/arduino/make-a-smart-automatic-pet-feeder-with-arduino-uno/	This website served as one of my references for my Final year Project. This project inspired me to do this project, However I have made upgrade and modifications to make it more efficient and reliable. Furthermore, I think this project is quite interesting because you can save a lot of time and energy to check on your beloved pets.
PIR Motion Sensor	Lady Ada	https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor	This Website served as one of my references for my Final Year Project. PIR sensor is one of my sensors that I use in my project, the function PIR is to detect motion of human and animal movement, so I think it's perfect for my project. Furthermore, they are small and low cost so you don't have to worry spend much money on the sensor
ESP8266 for IoT	Ida Hubschmann	https://tinyurl.com/2z2ka3wx	This Website served as one of my references for my Final Year Project. ESP 8266 is certainly vital for my FYP

			because Given its low cost, small size and adaptability with embedded devices, the ESP8266 is now used extensively across IoT devices. The ESP8266 module enables microcontrollers to connect to 2.4 GHz Wi-Fi, using IEEE 802.11 bgn. It can be used with ESP-AT firmware to provide Wi-Fi connectivity to external host MCUs, or it can be used as a self-sufficient MCU by running an RTOS-based SDK. The module has a full TCP/IP stack and provides the ability for data processing, reads and controls of GPIOs.

1.3 Project Description

Project justification:

The goal of this research is to discover a practical method for making significant time and labor savings.

Because people have other obligations, such as going to work in the morning and likely returning home at night, it is rather convenient and will make life easier for them.

Overall Function of the project:

In this project, the NodeMCU, ESP8266 serves as the primary controller. A servo motor opens and closes the feeding entry; an ultrasonic sensor

measures the amount of food; and a DHT-11 sensor measures the food's quality.

My project's primary goal is to regulate and keep an eye on the food that is given to my cat. The pet won't be hungry while waiting for me to get home because the PIR motion sensor will promptly pour out some food when it detects movement near the sensor.

The next Function Is using ultrasonic sensor to monitor quantity of food so that I will remember when to fill it up again, Finally DHT 11 will be used to detect humidity in the container so that the food will be in good condition.

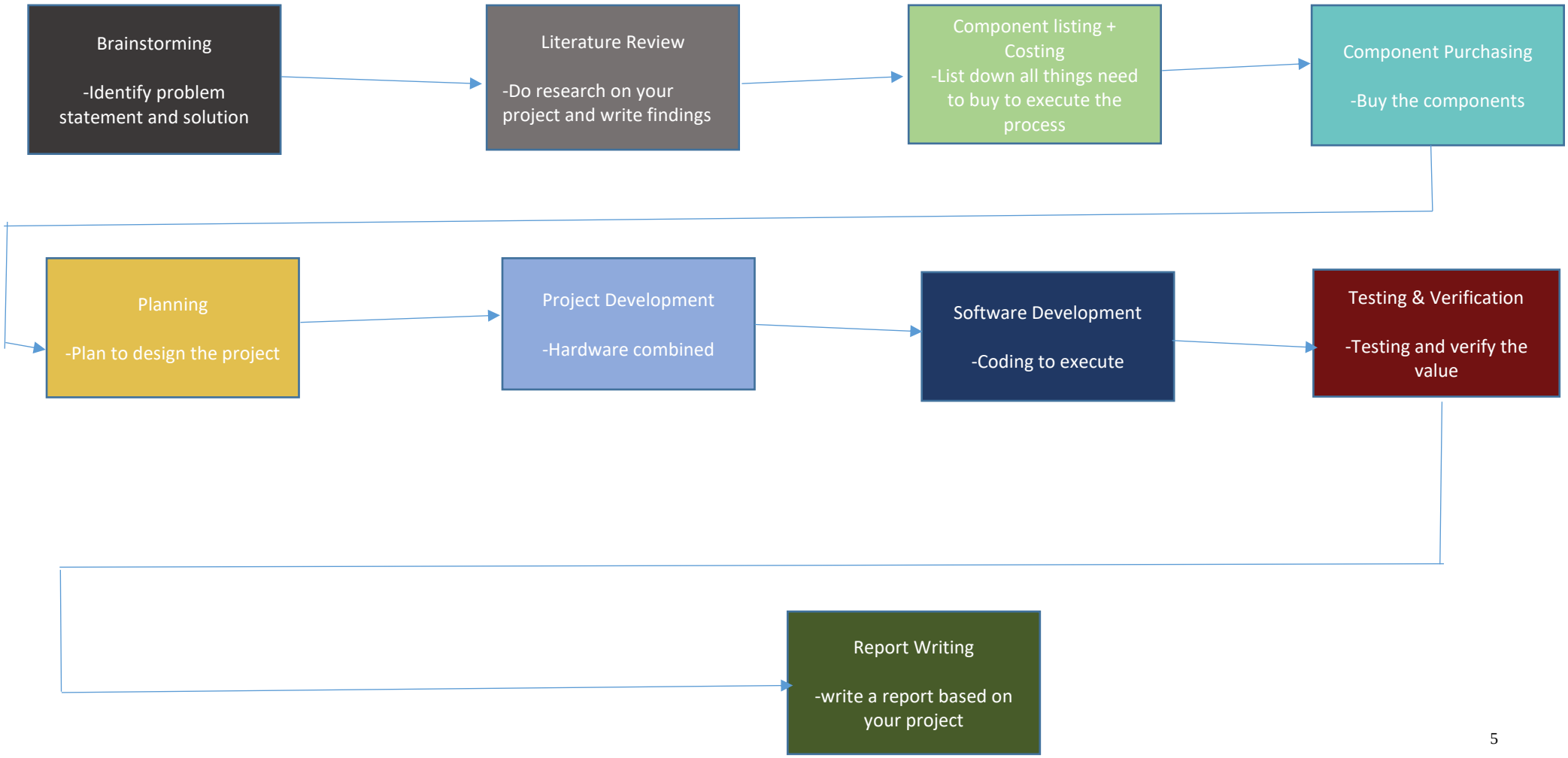
1.4 Project Objective

i. To design IoT smart feeder using the internet of things application that can monitor the quantity of food

ii. To build hardware that can display the sensor data on the dashboard

iii. To verify and test the smart feeder with its functionality so that food can be deliver to pet successfully

1.5 Scope of Works

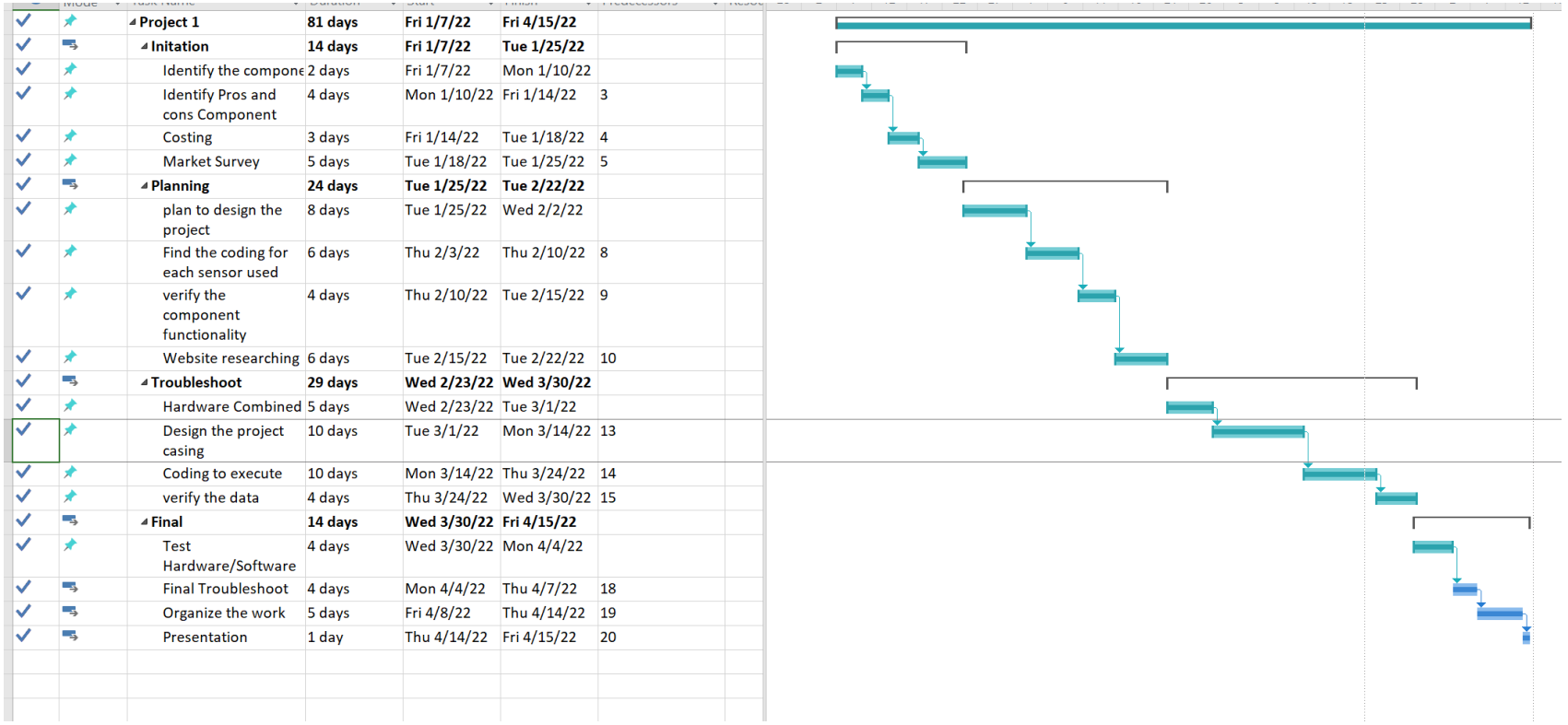


1.5 Project planning and monitoring

GANTT CHART

	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resourc
1		**Project 1**	**81 days**	**Fri 1/7/22**	**Fri 4/15/22**		
2		**Initation**	**14 days**	**Fri 1/7/22**	**Tue 1/25/22**		
3		Identify the compone	2 days	Fri 1/7/22	Mon 1/10/22		
4		Identify Pros and cons Component	4 days	Mon 1/10/22	Fri 1/14/22	3	
5		Costing	3 days	Fri 1/14/22	Tue 1/18/22	4	
6		Market Survey	5 days	Tue 1/18/22	Tue 1/25/22	5	
7		**Planning**	**24 days**	**Tue 1/25/22**	**Tue 2/22/22**		
8		plan to design the project	8 days	Tue 1/25/22	Wed 2/2/22		
9		Find the coding for each sensor used	6 days	Thu 2/3/22	Thu 2/10/22	8	
10		verify the component functionality	4 days	Thu 2/10/22	Tue 2/15/22	9	
11		Website researching	6 days	Tue 2/15/22	Tue 2/22/22	10	
12		**Troubleshoot**	**29 days**	**Wed 2/23/22**	**Wed 3/30/22**		
13		Hardware Combined	5 days	Wed 2/23/22	Tue 3/1/22		
14		Design the project casing	10 days	Tue 3/1/22	Mon 3/14/22	13	
15		Coding to execute	10 days	Mon 3/14/22	Thu 3/24/22	14	
16		verify the data	4 days	Thu 3/24/22	Wed 3/30/22	15	
17		**Final**	**14 days**	**Wed 3/30/22**	**Fri 4/15/22**		
18		Test Hardware/Software	4 days	Wed 3/30/22	Mon 4/4/22		
19		Final Troubleshoot	4 days	Mon 4/4/22	Thu 4/7/22	18	
20		Organize the work	5 days	Fri 4/8/22	Thu 4/14/22	19	
21		Presentation	1 day	Thu 4/14/22	Fri 4/15/22	20	

1.6.1 Timelines:



	Task Name	Duration	Start	Finish	Predecessors
✓	**Project 1**	**81 days**	**Fri 1/7/22**	**Fri 4/15/22**	
✓	**Initation**	**14 days**	**Fri 1/7/22**	**Tue 1/25/22**	
✓	Identify the compone	2 days	Fri 1/7/22	Mon 1/10/22	
✓	Identify Pros and cons Component	4 days	Mon 1/10/22	Fri 1/14/22	3
✓	Costing	3 days	Fri 1/14/22	Tue 1/18/22	4
✓	Market Survey	5 days	Tue 1/18/22	Tue 1/25/22	5
✓	**Planning**	**24 days**	**Tue 1/25/22**	**Tue 2/22/22**	
✓	plan to design the project	8 days	Tue 1/25/22	Wed 2/2/22	
✓	Find the coding for each sensor used	6 days	Thu 2/3/22	Thu 2/10/22	8
✓	verify the component functionality	4 days	Thu 2/10/22	Tue 2/15/22	9
✓	Website researching	6 days	Tue 2/15/22	Tue 2/22/22	10
✓	**Troubleshoot**	**29 days**	**Wed 2/23/22**	**Wed 3/30/22**	
✓	Hardware Combined	5 days	Wed 2/23/22	Tue 3/1/22	
✓	Design the project casing	10 days	Tue 3/1/22	Mon 3/14/22	13
✓	Coding to execute	10 days	Mon 3/14/22	Thu 3/24/22	14
✓	verify the data	4 days	Thu 3/24/22	Wed 3/30/22	15
✓	**Final**	**14 days**	**Wed 3/30/22**	**Fri 4/15/22**	
✓	Test Hardware/Software	4 days	Wed 3/30/22	Mon 4/4/22	
✓	Final Troubleshoot	4 days	Mon 4/4/22	Thu 4/7/22	18
✓	Organize the work	5 days	Fri 4/8/22	Thu 4/14/22	19
✓	Presentation	1 day	Thu 4/14/22	Fri 4/15/22	20

Gantt chart Description:

Initiation – In the first phase we have discussed among our group members of how the project is going to be in the long term. Firstly, we had identified the components that are going to use for our project, we also estimated the cost that going to cost us at the end of discussion.

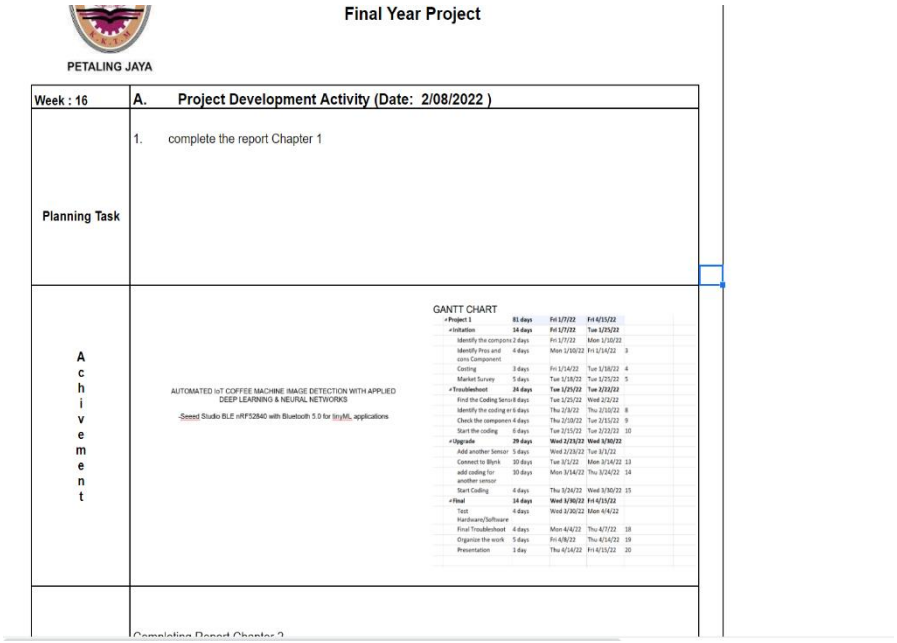
Planning – In the second phase we began to plan to execute the project for the design and hardware I stated in initiation earlier. and try to find some coding for each sensor used. In this phase it is to ensure the function of the sensor.

Troubleshoot – In the next phase is called Troubleshoot, we upgraded from one `sensor to two sensor and from two sensor we upgrade to three sensors, same goes `to actuators, we also troubleshoot the coding because it will not work if we keep `the same coding we made earlier. and we design the project casing

Final – In the Final Phase we finalize all the things we did earlier, we made last check for the function whether it is work or not, we also made sure the function will effectively and most importantly hygiene is secured, and the presentation come at the last place for us to show our hard work that we put in this project

1.6.2 Logbook Monitoring

Week : 16 (Week 1 Project 2)

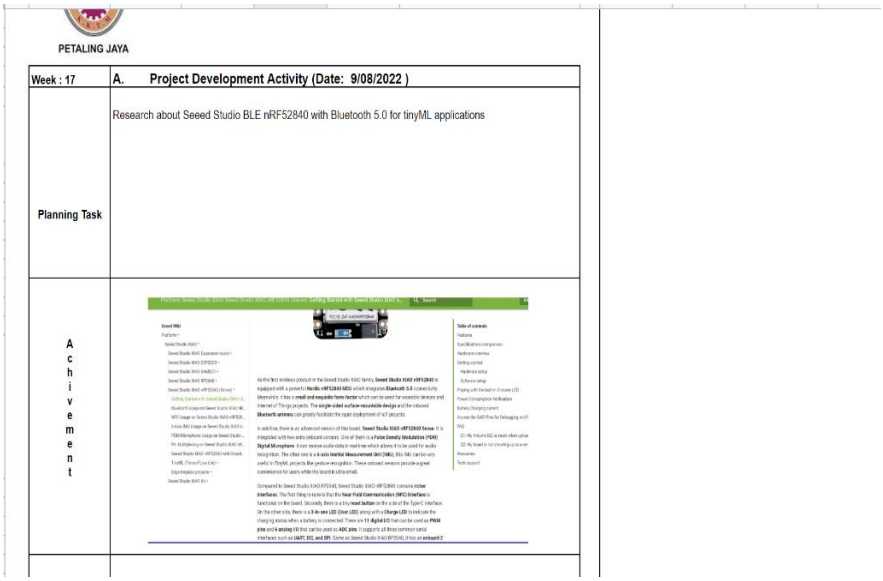


PETALING JAYA

Final Year Project

Week : 16	A. Project Development Activity (Date: 2/08/2022)
Planning Task	1. complete the report Chapter 1
Achivement	AUTOMATED IoT COFFEE MACHINE IMAGE DETECTION WITH APPLIED DEEP LEARNING & NEURAL NETWORKS -Seeed Studio BLE nRF52840 with Bluetooth 5.0 for tinyML applications GANTT CHART
	Completing Report Chapter 2

Week :17 (Week 2: Project 2)



PETALING JAYA

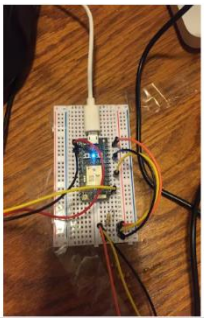
Week : 17	A. Project Development Activity (Date: 9/08/2022)
Planning Task	Research about Seeed Studio BLE nRF52840 with Bluetooth 5.0 for tinyML applications
Achivement	

Week :18 (Week 3: Project 2)

PETALING JAYA

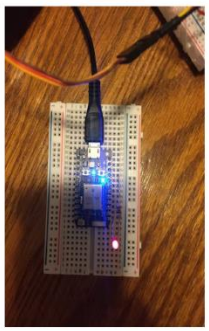
Project Progress Log

Final Year Project

Week : 18	A. Project Development Activity (Date: 16/08/2022)
Planning Task	Build a circuit and test it
Achivement	

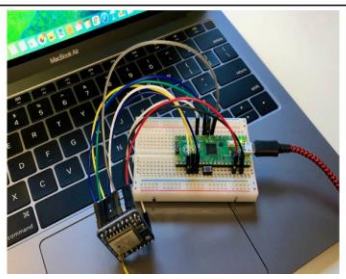
Week :19 (Week 4: Project 2)

PETALING JAYA

Week : 19	A. Project Development Activity (Date: 23/08/2022)
Planning Task	build another circuit and compare it
Achivement	

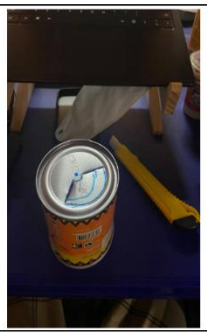
Week:20(Week 5: Project 2)

PETALING JAYA

Week : 24	A. Project Development Activity (Date: 30/08/2022)
Planning Task	test rasberry circuit
Achivement	

Week:21(Week 6: Project 2)

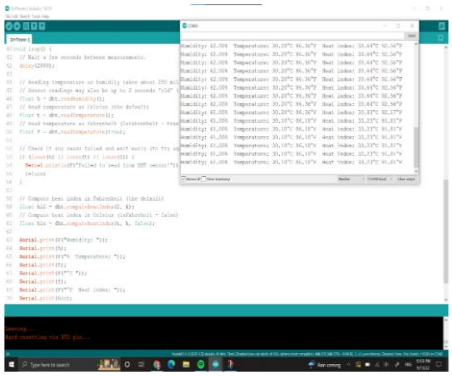
PETALING JAYA

Week : 26	A. Project Development Activity (Date: 13/09/2022)
Planning Task	Designing outer look of the project
Achivement	
	try combining the sensor coding

Week:22(Week 7: Project 2)

PETALING JAYA

Final Year Project

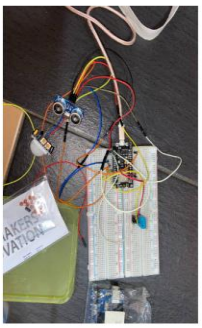
Week : 27	A. Project Development Activity (Date: 18/09/2022)
Planning Task	try combining the sensor coding
Achivement	

Week:23(Week 8: Project 2)

PETALING JAYA

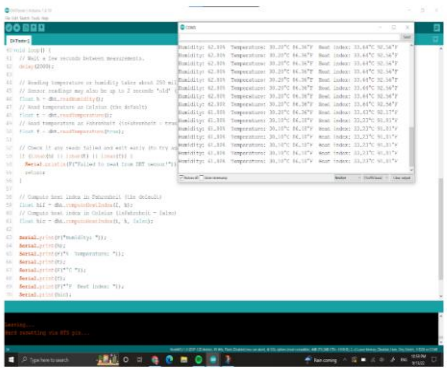
Week : 28	A. Project Development Activity (Date: 19/09/2022)
Planning Task	Decorate the Casing of the component
Achivement	
Next Task	Test the hardware circuit without servo motor

Week:24(Week 9: Project 2)

Week : 29	A. Project Development Activity (Date: 24/09/2022)
Planning Task	Test the hardware circuit without servo motor
Achivement	
Next Task	Verify the data is accurate or the opposite

Week 25(Week 10: Project 2)

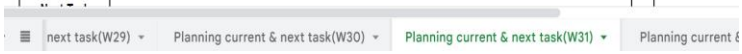
PETALING JAYA

Week : 30	A. Project Development Activity (Date: 25/09/2022)
Planning Task	Verify the data is accurate or the opposite
Achivement	
	Browse Electrical box for power supply

Week 25(Week 10: Project 2)

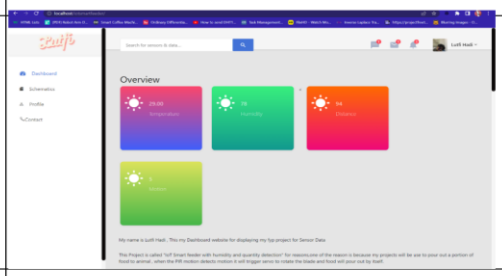
PETALING JAYA

Week : 31	A. Project Development Activity (Date: 3/10/2022)
Planning Task	Browse Electrical box for power supply
Achivement	
	Combine the hardware all together



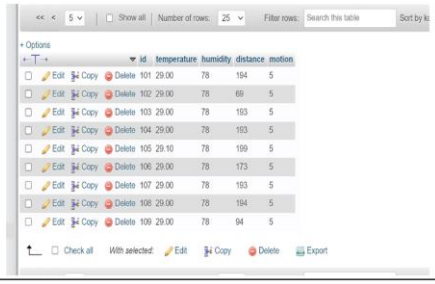
Week 26(Week 11: Project 2)

PETALING JAYA

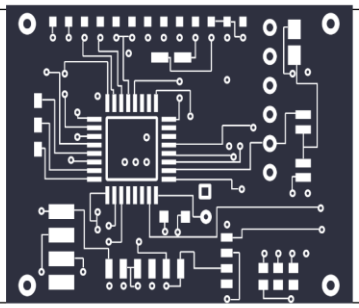
Week : 34	A. Project Development Activity (Date: 17/10/2022)
Planning Task	Send data to database and dashboard
Achivement	
Next Task	Verifiication

Week 27(Week 12: Project 2)

PETALING JAYA

Week : 35	A. Project Development Activity (Date: 18/10/2022)
Planning Task	Verification
Achivement	
Next Task	PCB design

Week 28(Week 13: Project 2)

Week : 36	A. Project Development Activity (Date: 24/10/2022)
Planning Task	PCB Design
Achivement	
Next Task	make presentation slides

Week 29(Week 14: Project 2)

PETALING JAYA

Week : 37	A. Project Development Activity (Date: 25/10/2022)
Planning Task	cutout design for final phase
Achivement	
Next Task	Change wifi SSID and Password in the coding

Week 30(Week 15: Project 2)

Week : 37	A. Project Development Activity (Date: 25/10/2022)
Planning Task	Change wifi SSID and Password in the coding
Achivement	34 const int echoPin = 14; 35 36 //wifi credential 37 38 const char* ssid = "Lutfi"; 39 const char* password ="Lutfilelouch00"; 40 const char* host = "172.20.10.2"; 41 42 43 void setup() {
Next Task	testing and verification again to double check abnormality

Week 31(Week 16: Project 2)

Week : 39	A. Project Development Activity (Date: 26/10/2022)
Planning Task	testing and verification again to double check abnormality
Achivement	
Next Task	cutoff extra excess of the product

Week 32(Week 17: Project 2)

PETALING JAYA

Week : 40	A. Project Development Activity (Date: 28/10/2022)
Planning Task	cutoff extra excess of the product
Achivement	
Next Task	prepare for presentation

CHAPTER 2
PROJECT DESIGN OVERVIEW

2.1 Product Design

This casing is for the esp8266 microcontroller. And for the big one is for ultrasonic sensor, pir sensor and DHT 11 but the main function of the big one is to pour pet food into the bowl

2.1.1 Front view

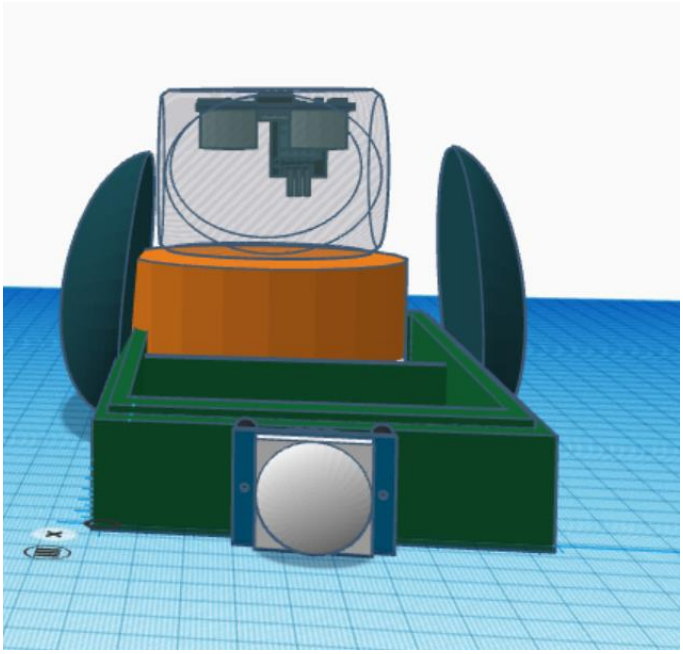


Figure 2.1.1: Front View

2.1.2 Top View

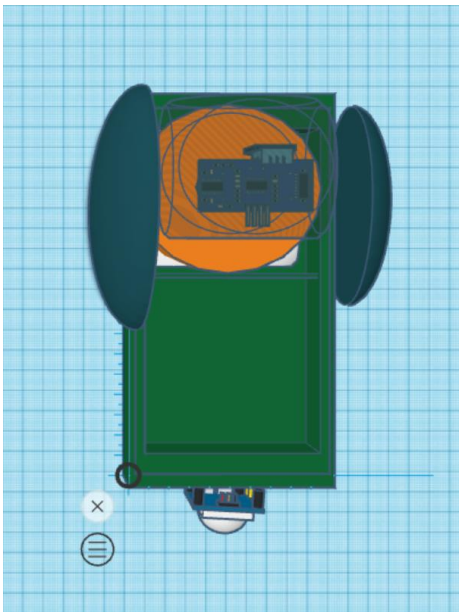


Figure 2.1.2: Top View

2.1.3 Side View

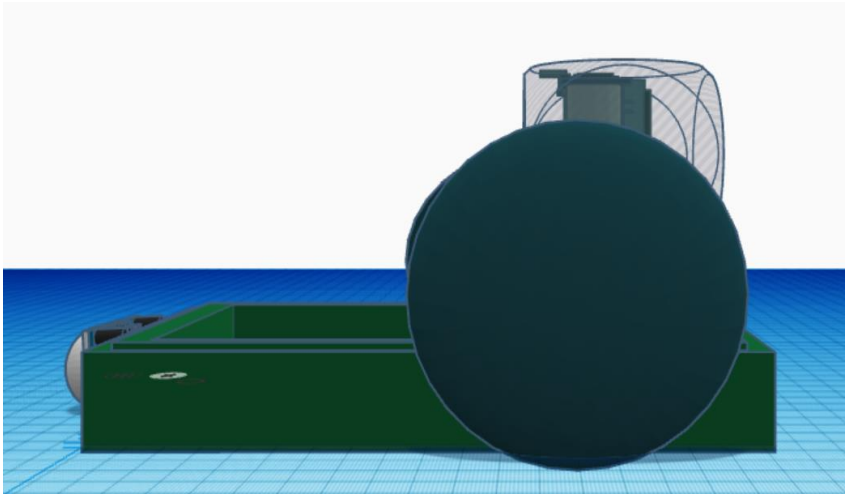


Figure 2.1.3: Side view

2.1.3 Isometric View

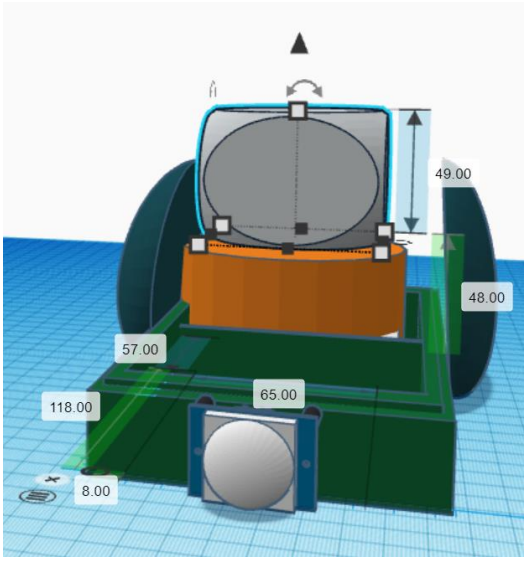


Figure 2.1.4: Isometric View

2.2 Product Specification

This section will explain the hardware that I used in my project. I will also explain the specification of the component such as size, accuracy and number of pins used.

Table 2.3: Product Specification

No	Name	Product	Specification
1	DHT11		• Low cost • 3 to 5V power and I/O • 2.5mA max current use during conversion (while

			requesting data) • Good for 20-80% humidity readings with 5% accuracy • Good for 0-50°C temperature readings ±2°C accuracy
2	Tower pro MG90s		• Operating Voltage: 4.8V to 6V (Typically 5V) • Stall Torque: 1.8 kg/cm (4.8V) • Max Stall Torque: 2.2 kg/cm (6V) • Operating speed is 0.1s/60° (4.8V) • Rotation: 0°-180°

3	ESP 8266		• Xtensa Single-core 32-bit L106 • 10-bit • Frequency 80MHz. • No Bluetooth
4	Ultrasonic Sensor		• Power Supply: +5V DC [5] • Quiescent Current: <2mA • Working Current: 15mA • Ranging Distance: 2cm – 400 cm/1" – 13ft • Effectual Angle: <15°
5	PIR sensor		• 3.3V - 5V input voltage • up to 20 feet (6 meters) 110 degrees x 70 degrees detection range • Rectangular

2.3 Block Diagram

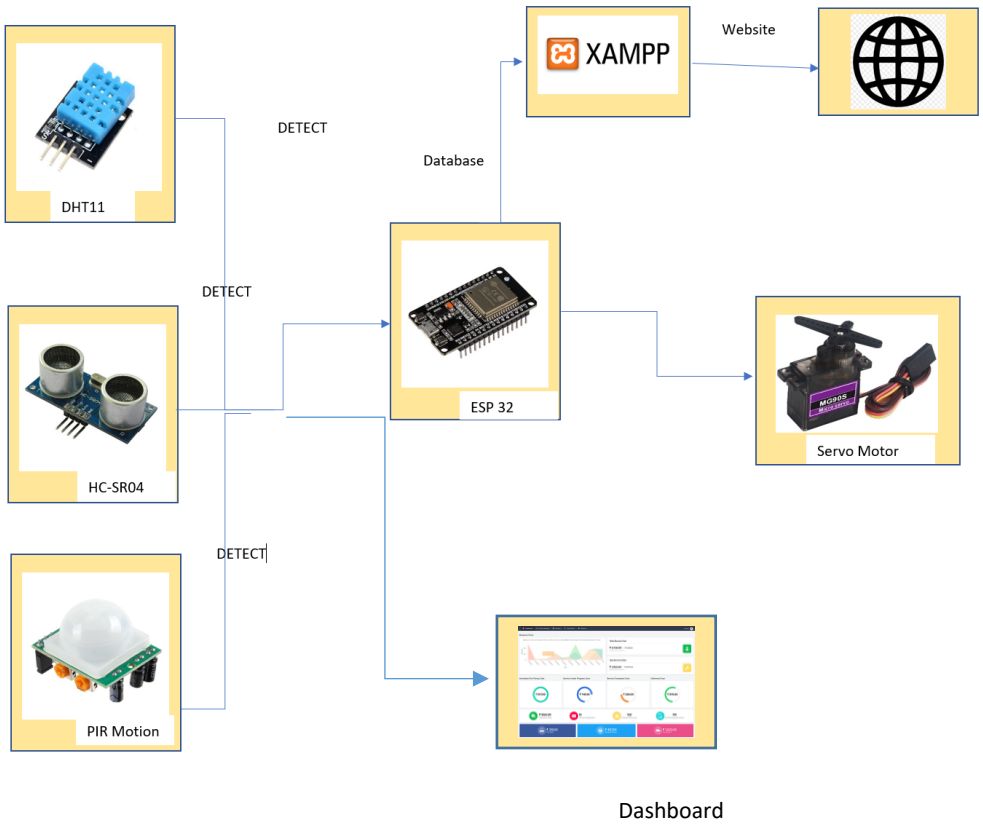


Figure 2.3.0: Block Diagram

This Block Diagram shows the flow of how the system works in the project. By using 3 main sensors as the input and 1 output for the project. The data will be sent to the microcontroller ESP32[9]. It will send the data to the database XAMPP, so the data will be save by Cloud storage. The data will be display on the website as dashboard for the end user to see the result of readings

Input

1. **DHT 11**
2. **HC-SR04**
3. **PIR MOTION**

Output

1. **Servo motor**
2. **Database**
3. **Dashboard**

Microcontroller

1. **ESP 8266**

2.4 Product Cost

Table 2.4: Cost of the Product

No	Product	Unit	Cost (RM)
1	DHT 11	1	4.90
2	Servo Motor(MG90s)	1	16.90
3	ESP8266	1	15.00
4	Ultrasonic Sensor	1	6.00
5	PIR Sensor	1	5.00

6	Container	2	50.00
7	Casing (Project) 63x58x35mm	1	20.00
8	Adapter wire	1	10.00
9	Jumper Wire	50	5.00
10	PCB for ESP8266	1	32.00
Total Price (RM)			163.80

To make the circuit's wiring simple to understand, Fritzing software was used to create it. PIR's signal pin was placed at position D7 on the ESP-32, while Ultrasonic's trigger and echo pins were placed at positions D6 and D5, respectively, and DHT11 was placed at position D4.

2.5 Electronic Schematic Circuit

To make the circuit's wiring simple to understand, Fritzing software was used to create it. PIR's signal pin was placed at position D7 on the ESP-32, while Ultrasonic's trigger and echo pins were placed at positions D6 and D5, respectively, and DHT11 was placed at position D4.

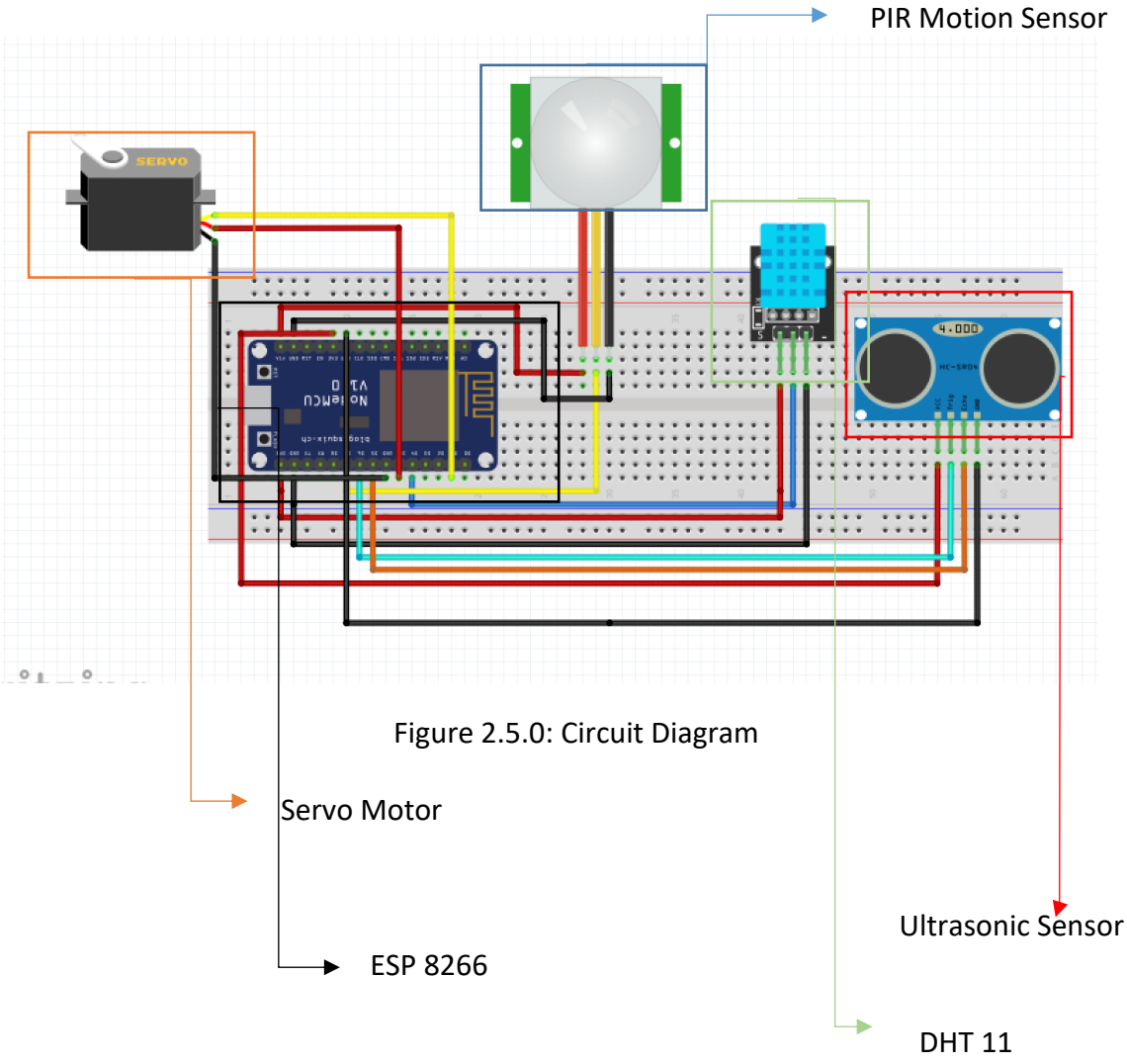


Figure 2.5.0: Circuit Diagram

CHAPTER 3
PRODUCT DEVELOPMENT

3.1 Introduction

This part will go into detail on how my project's hardware and software were developed. It also covers circuit construction, which I combined with the other components during testing and verification. For hardware development will explain the function of the sensors and will cover step by step the process in the making of the system. Same goes for the Software development show all the step from initial to the end.

3.2 Circuit Construction

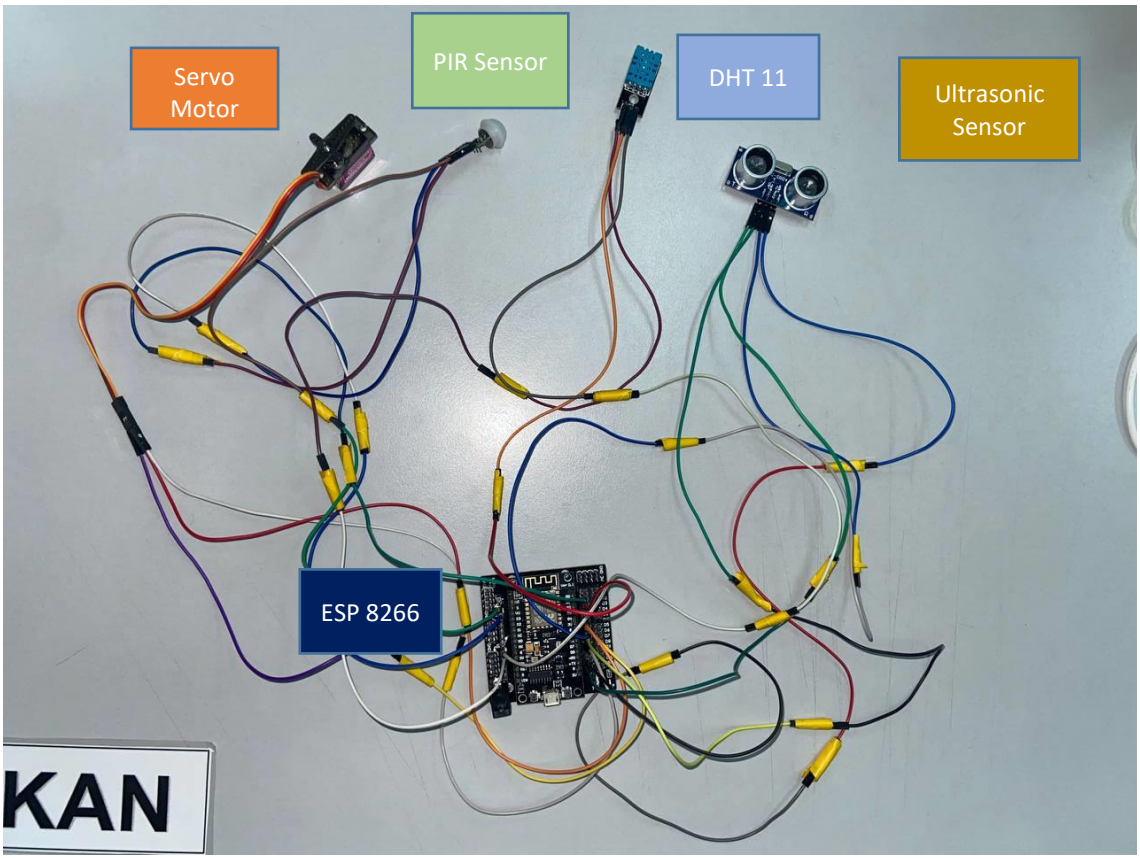


Figure 3.0.0: Circuit Construction

The IoT Smart Feeder with Food Quantity Monitoring is the title of my final year project, and this is the circuit I built for it. I used an ESP8266 microcontroller to connect to Wi-Fi and transfer the sensor reading data to the cloud. I used an ultrasonic sensor to keep track of the food's quantity, and a DHT-11 to keep track of the food's humidity. I utilized a PIR sensor anytime motion was detected to cause the servo motor to turn on and for the food to pour up.

Circuit Diagram

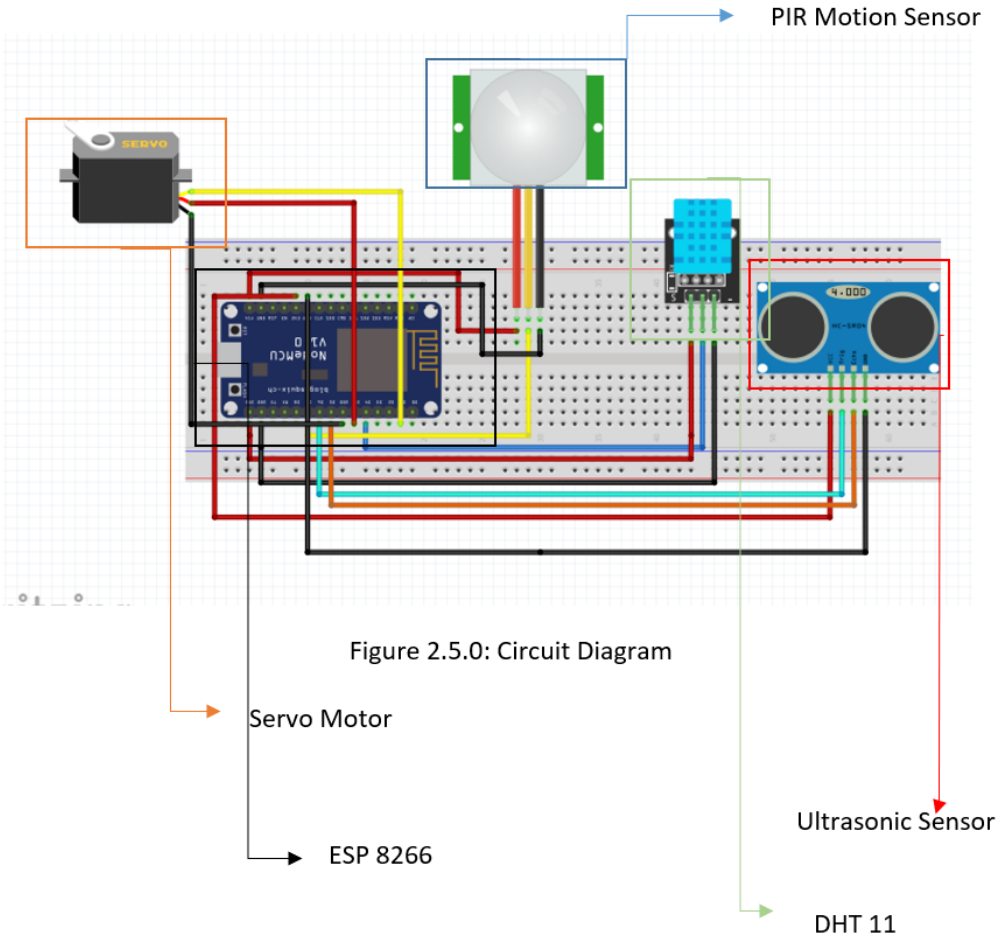


Figure 2.5.0: Circuit Diagram

3.3 Hardware Development

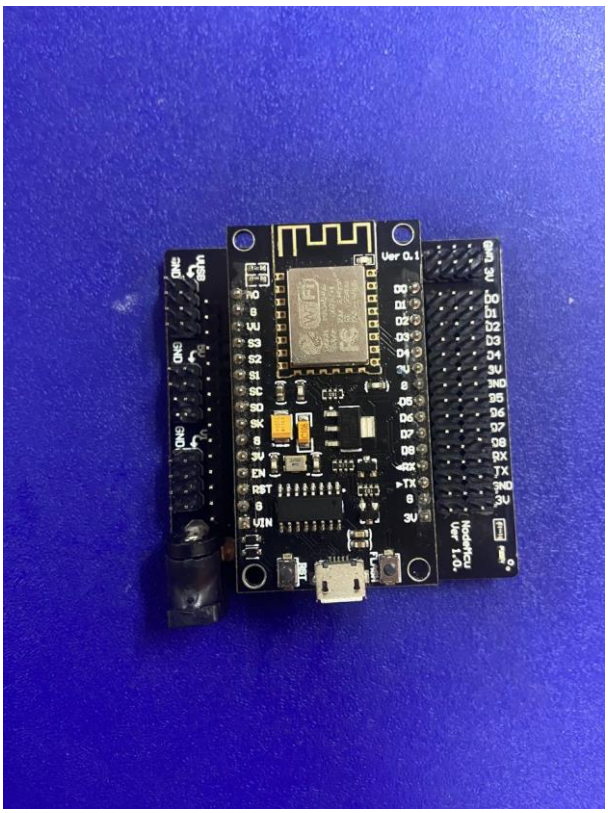


Figure 3.1.0: PCB with ESP 8266

The PCB I used for this project is the initial hardware development. It can supply my ESP8266 controller with power at up to 16V. Additionally, it contains a lot of pins, so you won't have to worry about running out.

The PCB I used for this project is the initial hardware development. It can supply my ESP8266 controller with power at up to 16V. Additionally, it contains a lot of pins, so you won't have to worry about running out of pins.

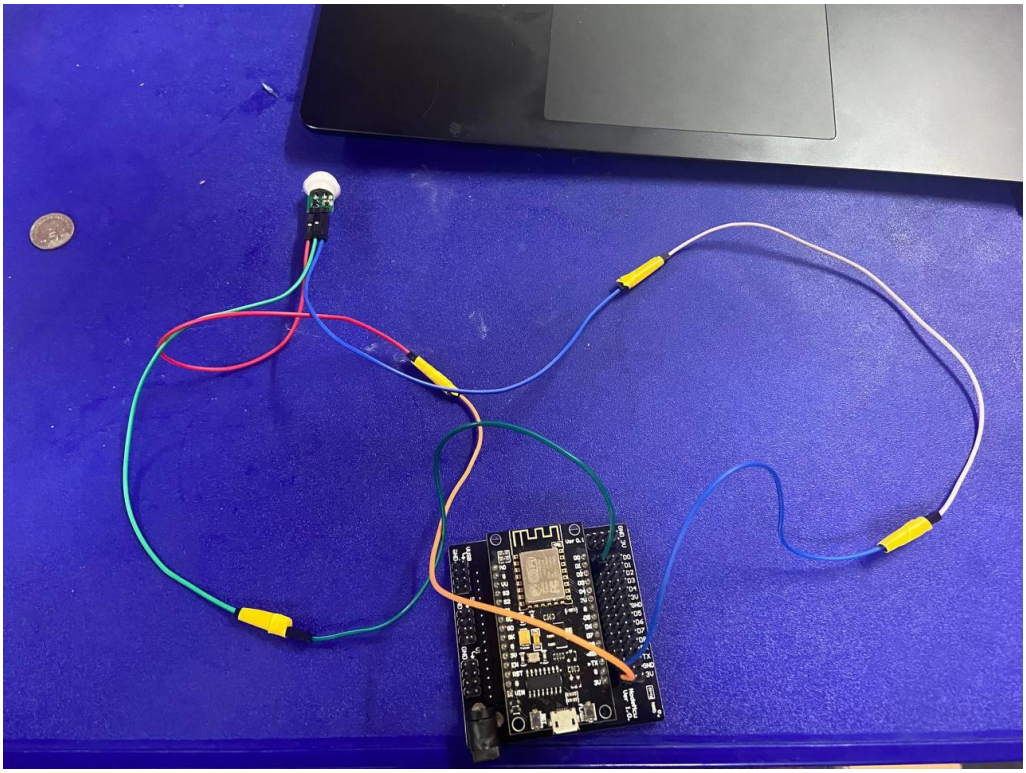


Figure 3.1.1: ESP 8266 with PIR

In this section I have started with one sensor and act as my first input

The pin data connected with this PIR is D1 which GPIO 5 in ESP 12-E library VCC connect with 3V and GND connect to the GND.PIR sensor will detect motion of the pet nearby.

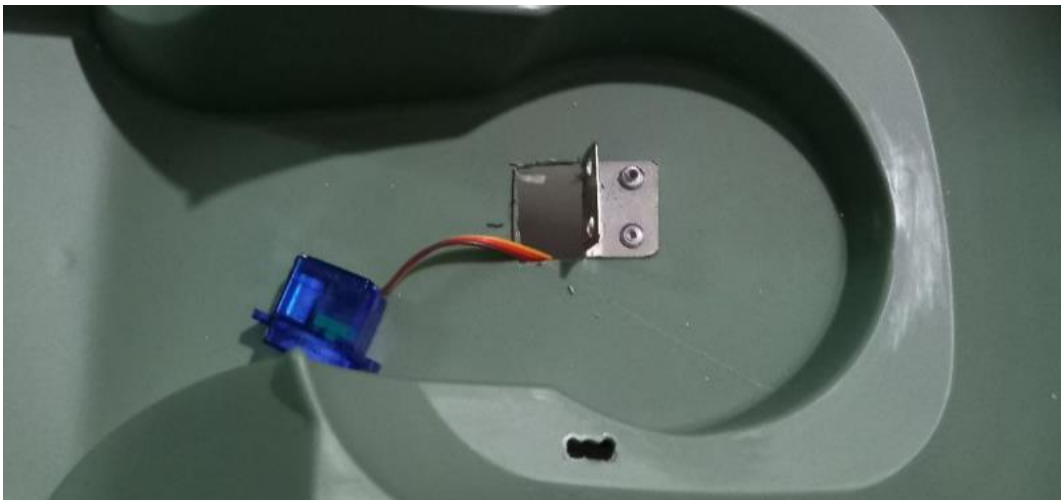


Figure 3.1.2: Servo Motor case

This servo I have design the product casing to look amazing and tidy. The servo motor that I use is MG90s, I have connect to D8 which is GPIO15, VCC to 3v3.

Figure 3.1.3: Bottom view of the servo

This is the bottom view of the servo motor I attach in container to make it spin and eventually pours up one amount of food

.

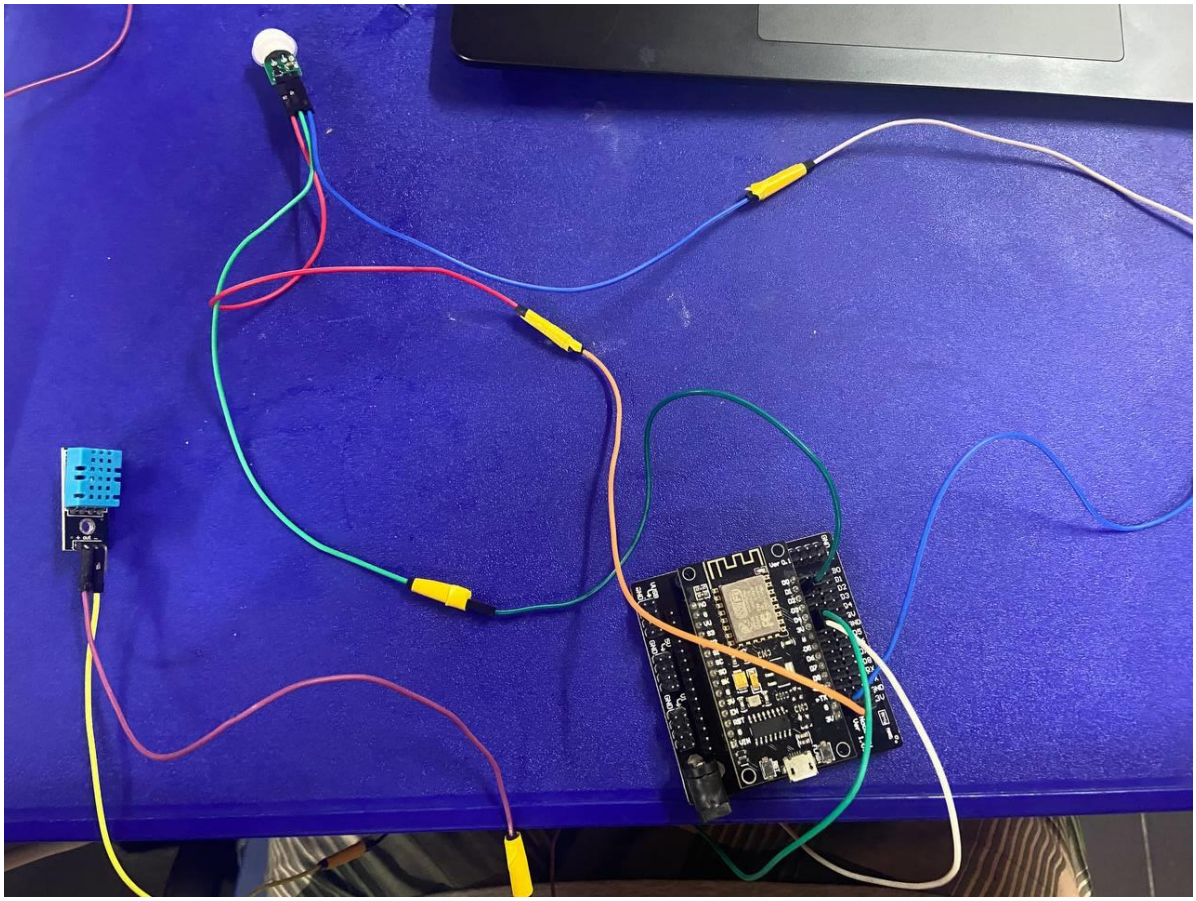


Figure 3.1.4: ESP 8266 + DHT 11 + PIR

In this phase I have added one more sensor to detect two status which is Temperature and humidity although it is ultra-low cost sensor so it will not accurate

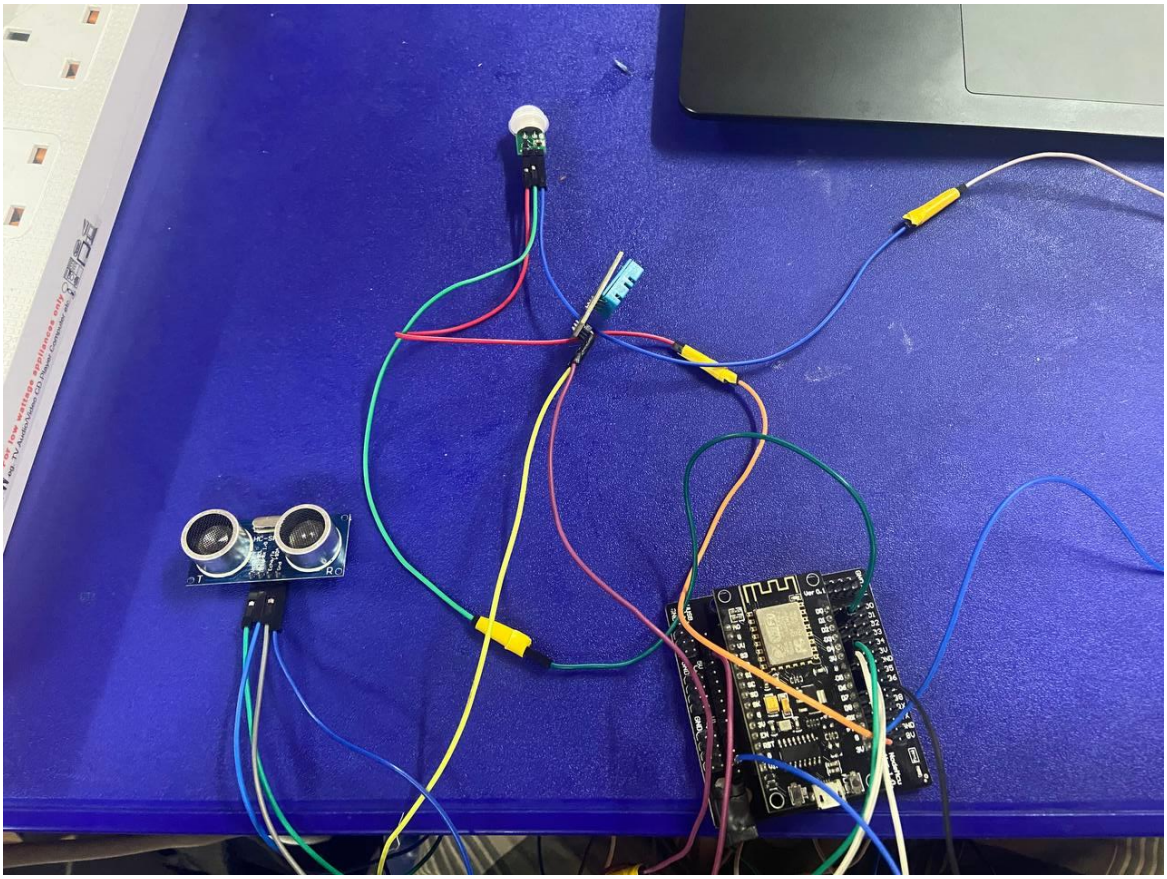


Figure 3.1.5: ESP 8266 + Ultrasonic sensor + DHT 11 + PIR

This is the third sensor, which is ultrasonic, we can monitor the distance, but alternatively we can measure food quantity level with this sensor, Ultrasonic sensor placement is in bottom of the lid

Figure 3.1.6: All components combined

Lastly, all components are combined together including servo motor as the main output, Ultrasonic Data output trigger is placed on D6 while echo is in D5. DHT 11 data output is placed D4. Pir sensor data output is place on D1 and lastly the servo is placed on D8. Combined it together I have finished the hardware of IoT smart feeder.

Table 2.5: Function of components

No	Name	Product	Function
1	DHT11		It can detect **temperature** and **humidity** of the food so that the food will be in great condition
2	Tower pro MG90s		When a motion sensor detects motion, this servo will rotate, rotating the containers until the food spills out of the hole.
3	ESP 8266		The function of this microcontroller is to connect the sensors, and let the data transfer send the data over Wi-fi.
4	Ultrasonic Sensor		It can detect the distance between food from the lid of the container, so that we know the quantity of the food left in the container.
5	PIR sensor		It will detect cat's movement through motion and it will trigger the output which the servo motor, the food will come out as by itself

PCB Layout

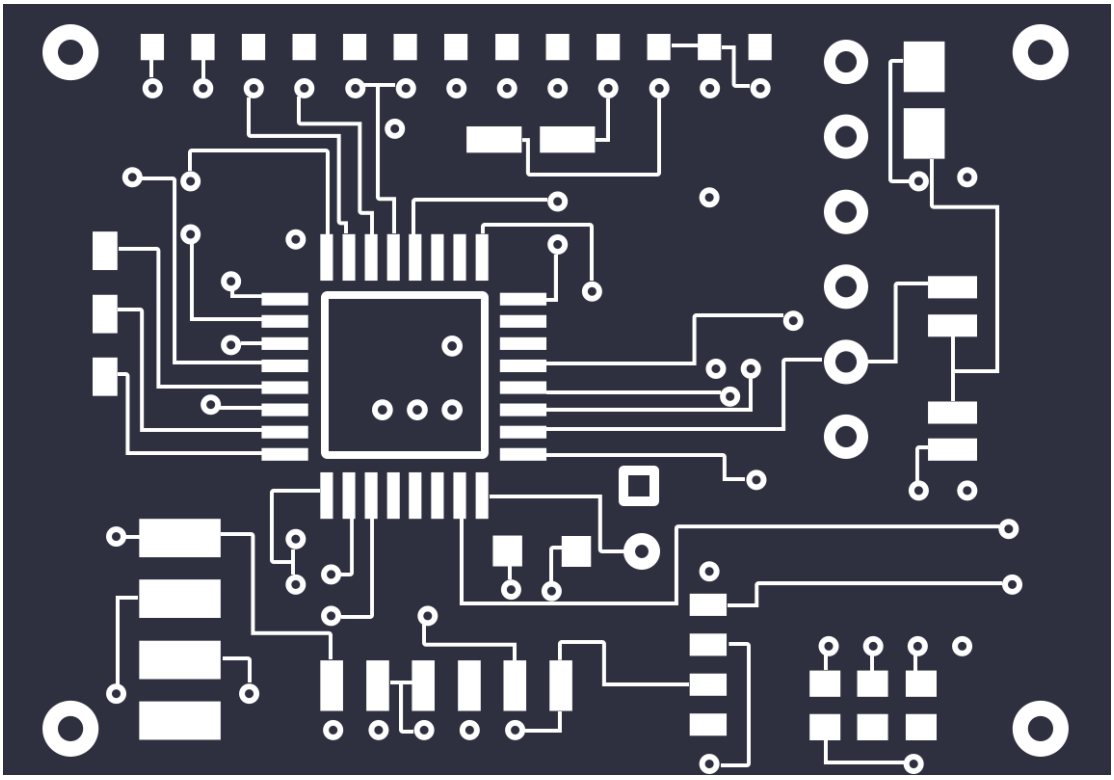


Figure 3.1.7: PCB layout

This is my PCB for my Final Year Project IoT Smart Feeder I have added power supply 6v to connect my Servo motor in this piece of PCB. I have also included DHT 11 in this PCB to monitor the temperature and humidity.

PCB Schematics

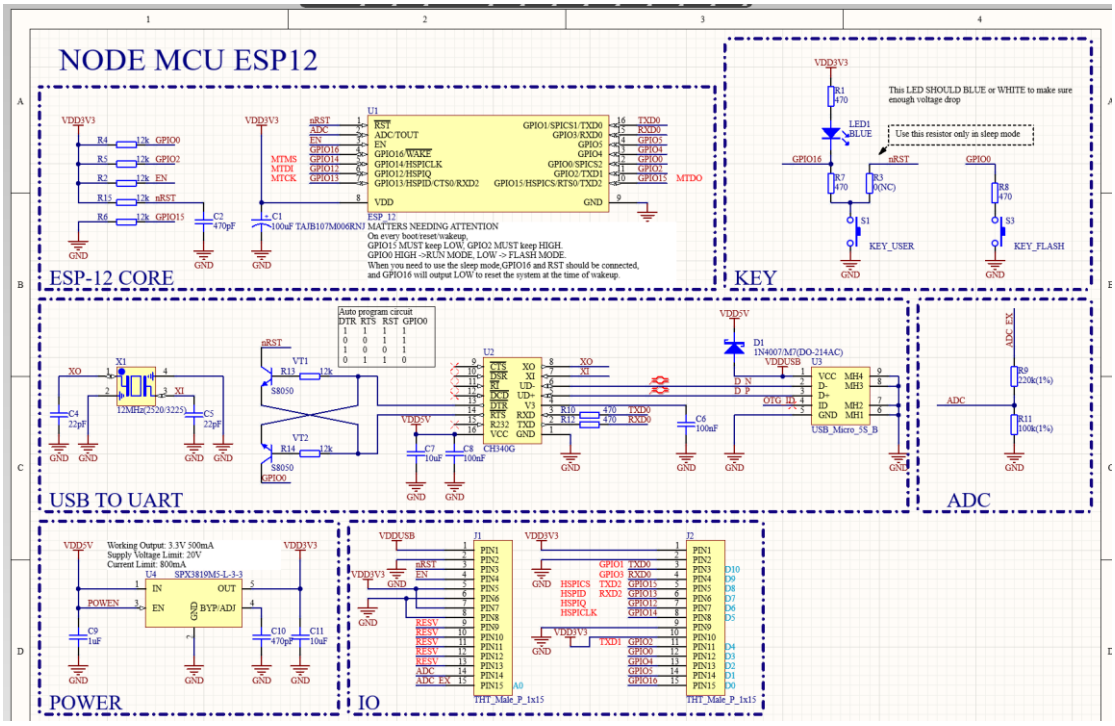


Figure 3.1.8: PCB schematic

This is my PCB Schematic, I have use ESP 8266 as my controller. I'm using DHT 11 to monitor the quality of the food through temperature and humidity, I have also added power supply so that I can connect my servo motor powered with 5V.

3.4 Software Development

Dashboard Website

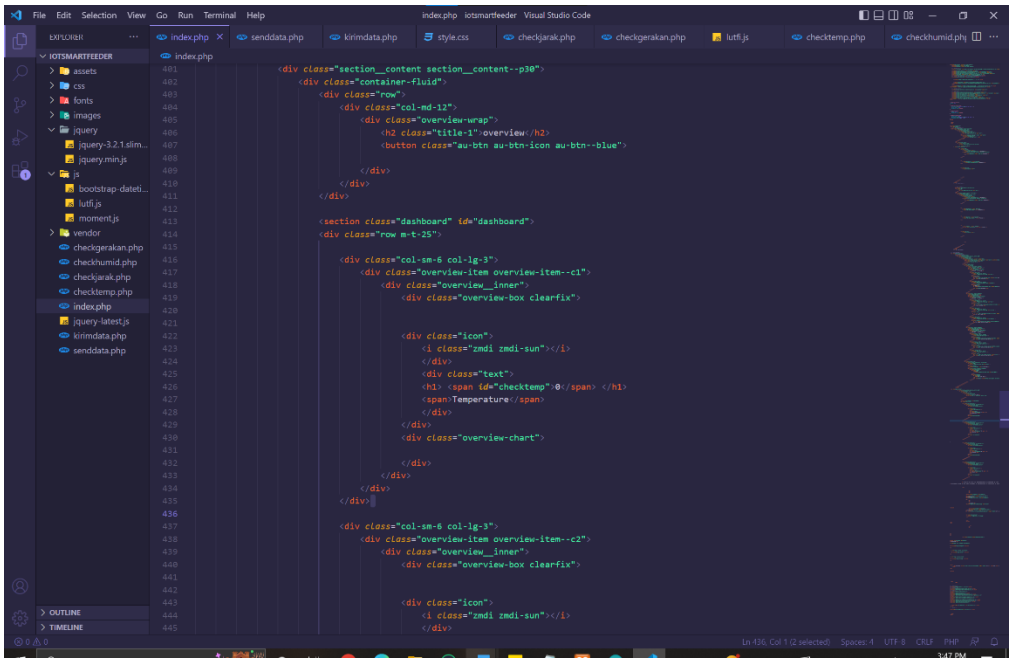


Figure 3.4.0: Coding for Dashboard website

This is the coding for Dashboard Website, I used Visual Studio Code for the editor. Programming language I have used in this section is HTML, CSS, JavaScript and PHP. HTML is for structure of the website, CSS is for design of the website. JavaScript is what make website unique because of the smooth scrolling. PHP language is to send data to the database.

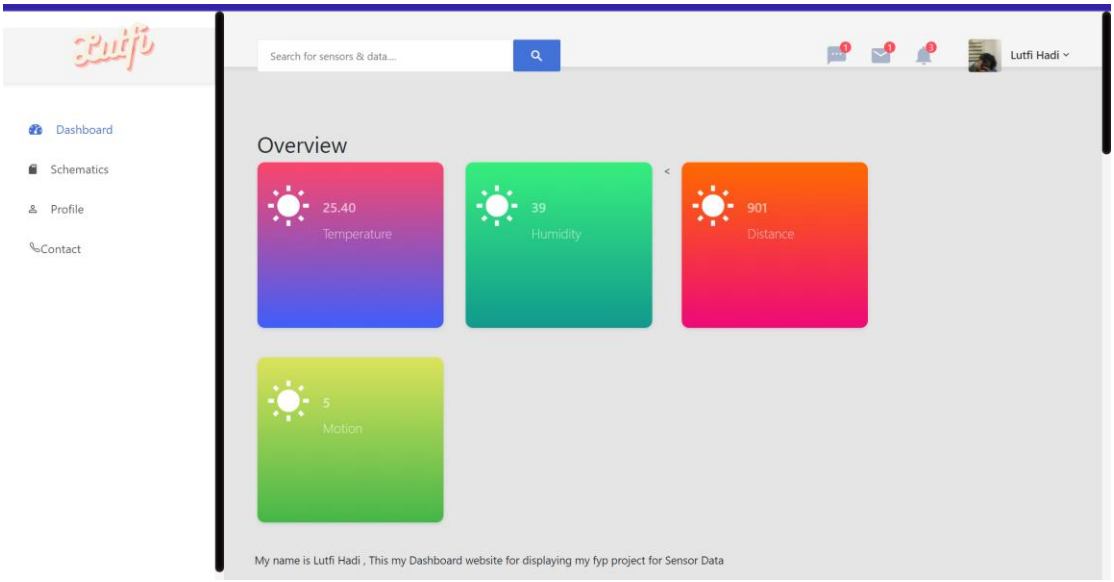


Figure 3.4.1: Website Dashboard

This is my Web dashboard for my Final Year Project, it is use to display the sensor reading so we can monitor it with our own devices. This is one of the IoT implementation in this phase, it also acts as Front end for my website Dashboard.

Database Data Storing

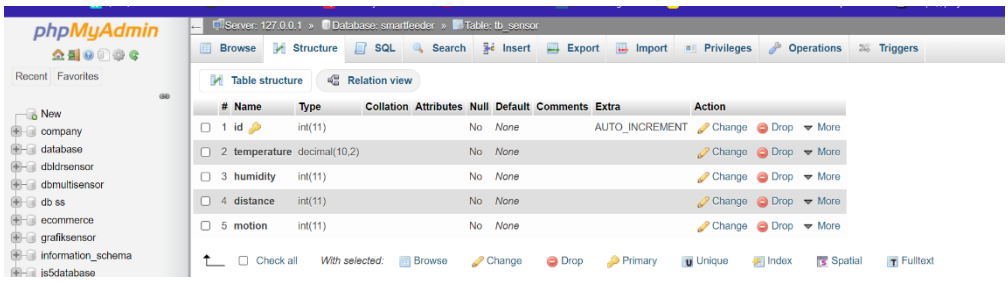


Figure 3.4.2: Table sensor of Database

This is my table sensor for database, this table is structured and act as the title of the sensor reading, from Visual Studio code coding PHP it will link up to Database resulting Sensor reading from Dashboard syncs with the Database.

		id	temperature	humidity	distance	motion
☐	Edit Copy Delete	1	60.00	70	10	1
☐	Edit Copy Delete	2	32.00	54	100	100
☐	Edit Copy Delete	3	50.00	81	100	1
☐	Edit Copy Delete	4	25.30	39	1207	5
☐	Edit Copy Delete	5	25.30	39	1197	5
☐	Edit Copy Delete	6	25.30	39	1197	5
☐	Edit Copy Delete	7	25.30	39	1197	5
☐	Edit Copy Delete	8	25.30	39	1197	5
☐	Edit Copy Delete	9	25.30	39	1197	5
☐	Edit Copy Delete	10	25.30	39	1198	5
☐	Edit Copy Delete	11	25.30	39	1199	5
☐	Edit Copy Delete	12	25.30	39	1199	5
☐	Edit Copy Delete	13	25.30	39	1202	5
☐	Edit Copy Delete	14	25.30	39	1197	5
☐	Edit Copy Delete	15	25.30	39	1197	5
☐	Edit Copy Delete	16	25.40	39	1197	5
☐	Edit Copy Delete	17	25.40	39	1197	5
☐	Edit Copy Delete	18	25.40	39	1196	5
☐	Edit Copy Delete	19	25.40	39	1199	5

Figure 3.4.3: Database

This is sensor reading captured in my database, I have displayed the reading to dashboard so that people can monitor it wirelessly, I have also has added the Database to store the past data reading so we can make a statistic based on data that we stored. This is one of the implementations of IoT for my Final Year Project.

Figure 3.4.4: Arduino IDE

This Arduino IDE uses C++ coding language, I am able to combine all the coding from each sensor and actuators to become one. I have also combined coding of the Wi-Fi and coding to display to the dashboard and then it will store the data to the database.

Block Diagram (Software)

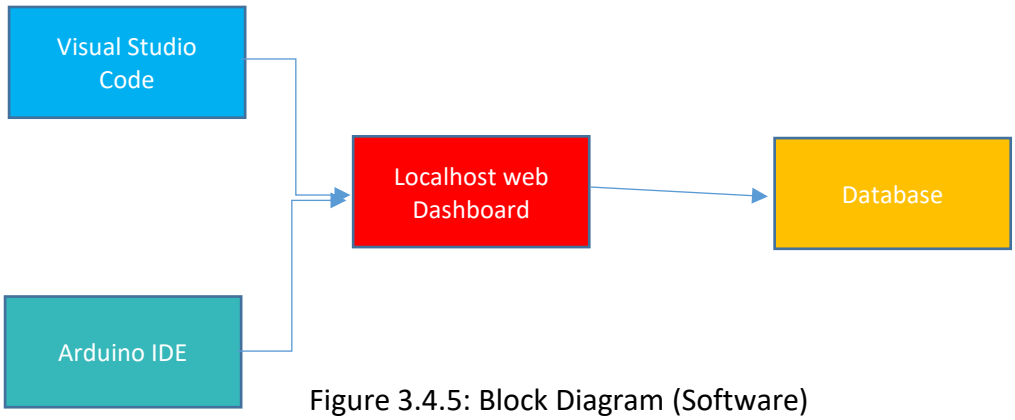


Figure 3.4.5: Block Diagram (Software)

PCB/SCHEMATICS SOFTWARE

Circuit Schematics (Fritzing)

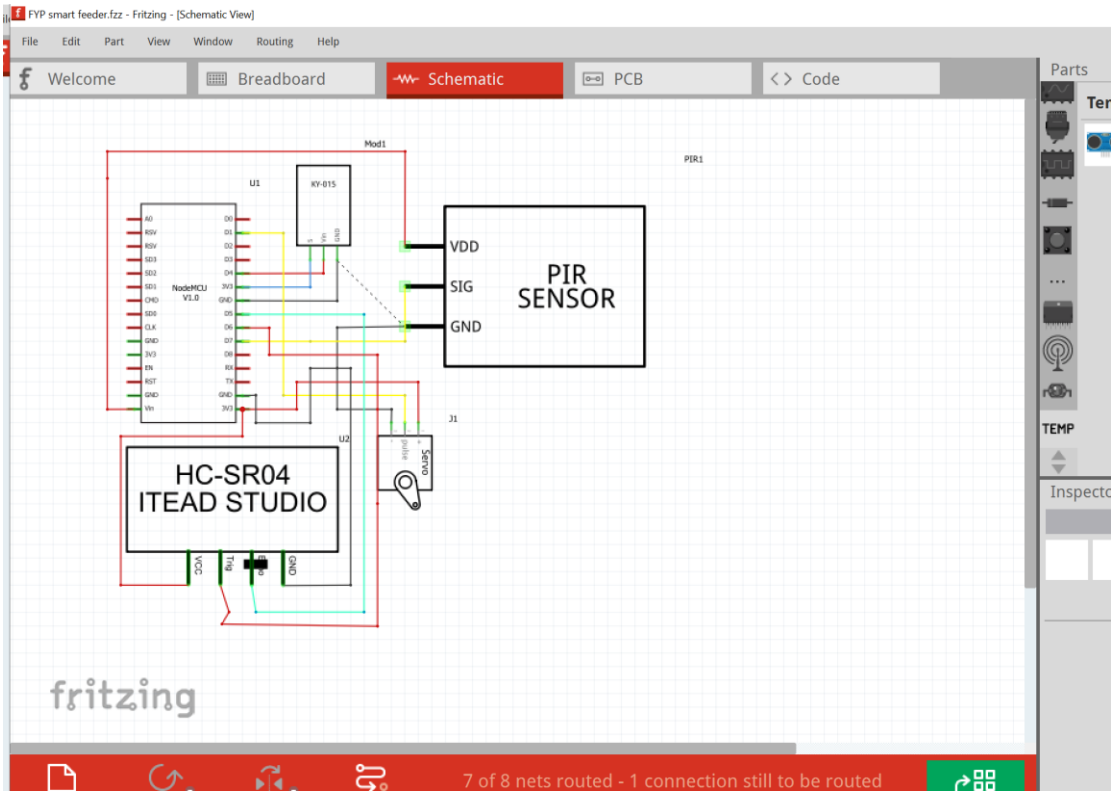


Figure 3.4.6: Circuit Schematics (Fritzing)

The software I used to create schematics design above is Fritzing, it is very convenient software to build circuits. One of the best things is the community constantly update the components so that the consumer can use.

PCB(Software)

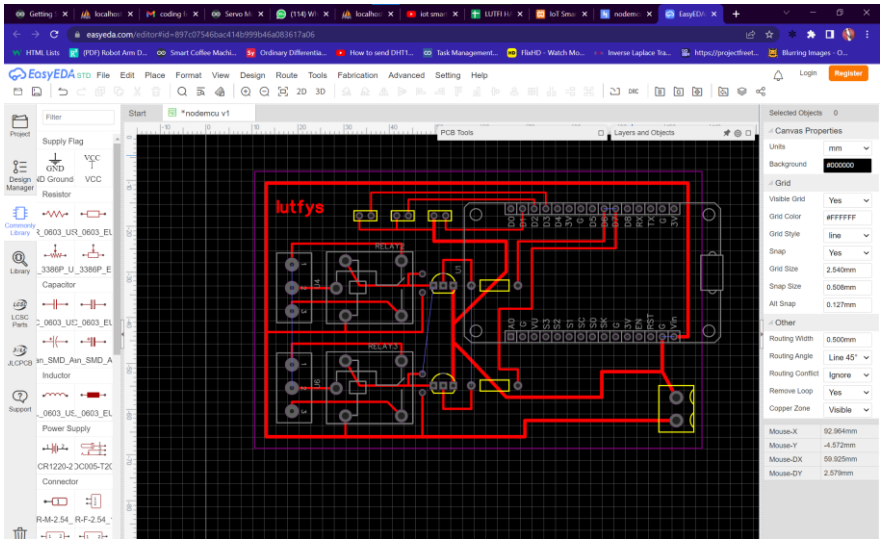


Figure 3.4.7: PCB(EasyEDA)

For PCB I used EasyEDA software, it’s quite easy to use than Altium Designer. Altium is heavier in terms of size capacity. EasyEDA also offer to publish online so that the community can use for their own project.

3.5 Testing and Verification

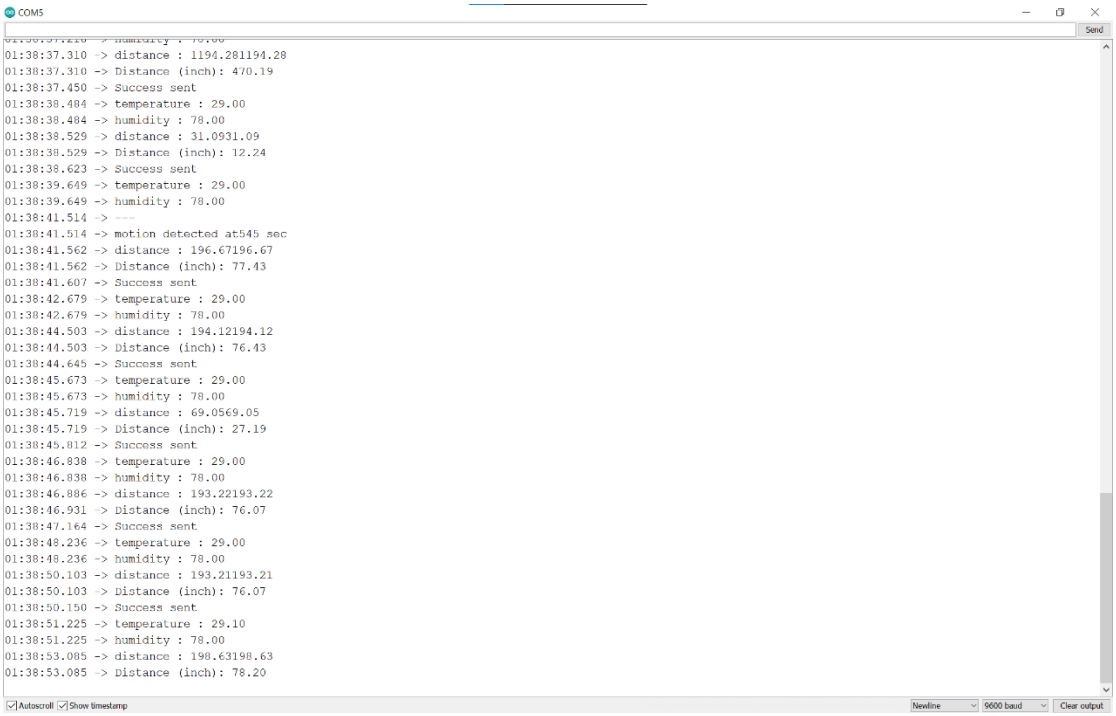


Figure 3.5.1: Serial Monitor of the output

In this phase I have to test the hardware I had combine earlier to connect to the database and display the reading to the website dashboard. As you can see my sensor reading “Success sent!” which means the sensor successfully store in database and successfully display in website dashboard.

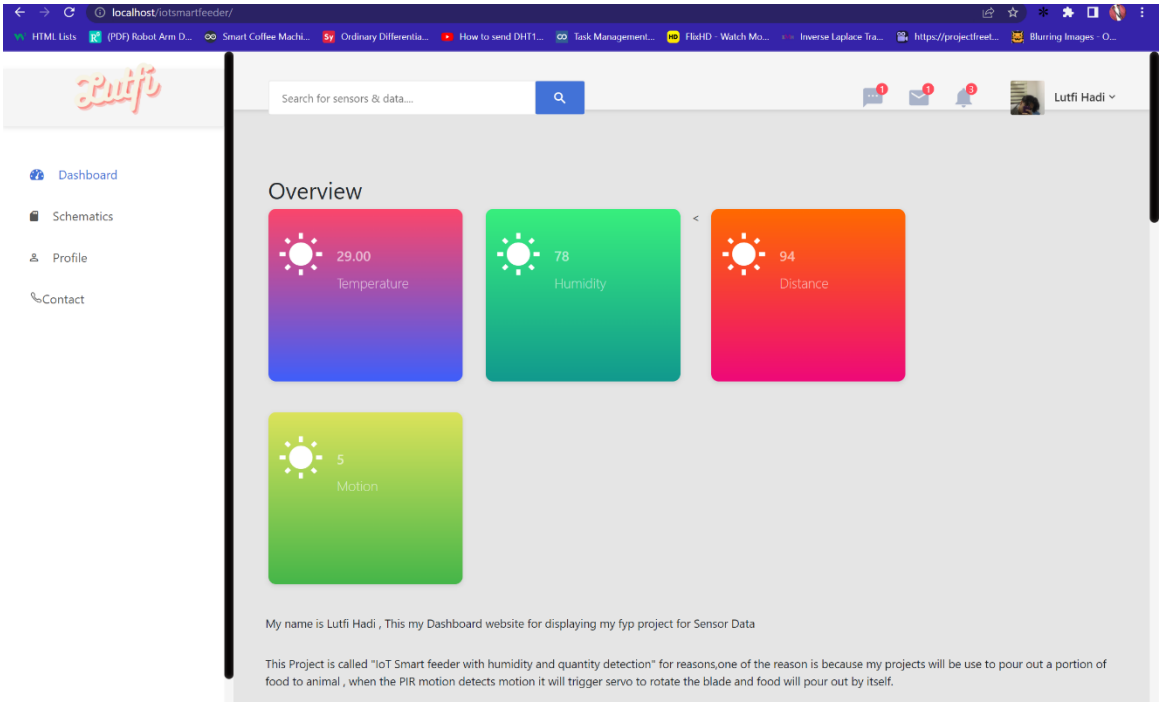


Figure 3.5.2: Dashboard live sensor reading

This is the same phase when I connect the Wi-Fi to ESP 8266, this is possible because the coding in the visual studio code with languages of HTML, CSS, JavaScript and PHP.

This website can be access by anyone once I start hosting this as a website.

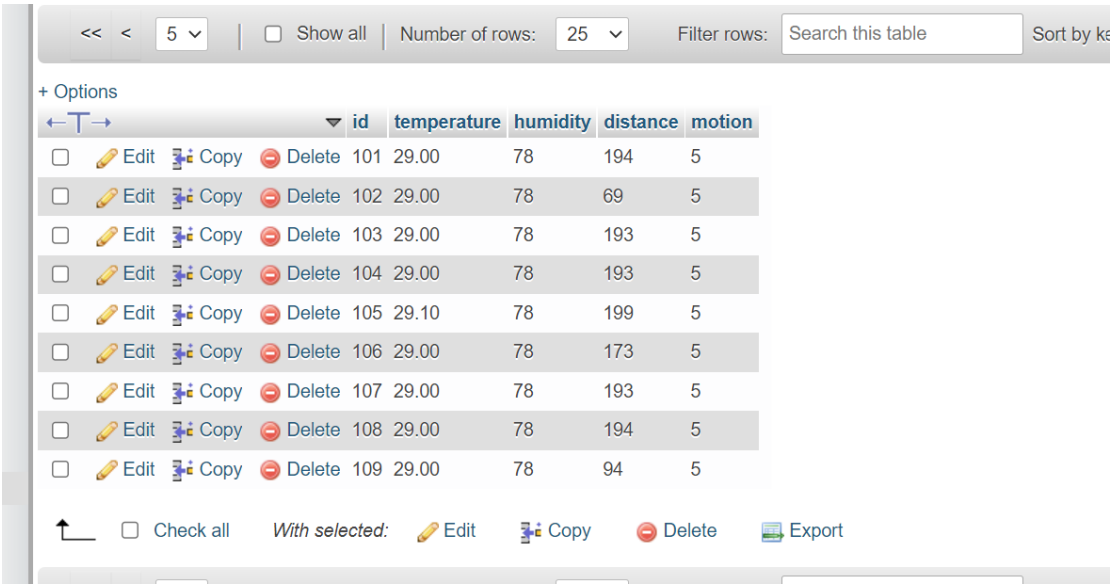


Figure 3.5.3: Database sensor reading

This is Database sensor reading, it will store the data, based sensor reading. It's secure place to store the data that coming over the time. I used PhpMyAdmin as my main Database storing.

CHAPTER 4
DATA, RESULT, ANALYSIS & TROUBLESHOOTING

4.1 Introduction

This chapter will go in detail about Data, result, analysis and Troubleshooting. The Data and result we got from sensor should be documentation in this report so that we know the statistical data of the project from the statistical data we can do Analysis to make a conclusion.

4.2 Data

Sensor	Parameters	Data
DHT 11	• Good for 20-80% humidity readings with 5% accuracy • Good for 0-50°C temperature readings ±2°C accuracy	Average humidity data captured is 56. means it's in between.
Ultrasonic	• Operating speed is 0.1s/60° (4.8V) • Max Stall Torque: 2.2 kg/cm (6V)	The data of distance captured depends on quantity of food, the more quantity of food the less the distance.
PIR Sensor	• up to 20 feet (6 meters) 110 degrees x 70 degrees detection range	The data of motion captured depends on motion detected during testing & verification

Table 4.2: Sensor Data

4.3 Result

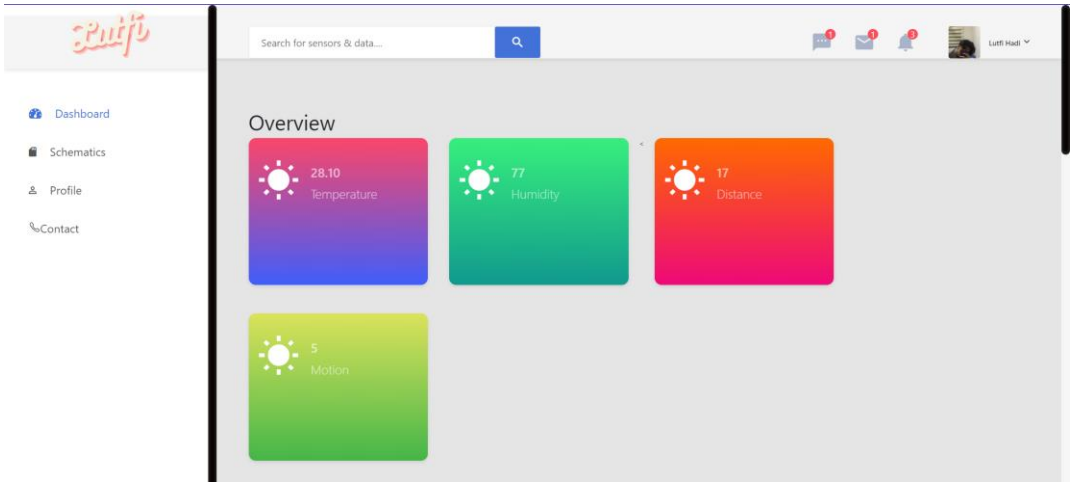


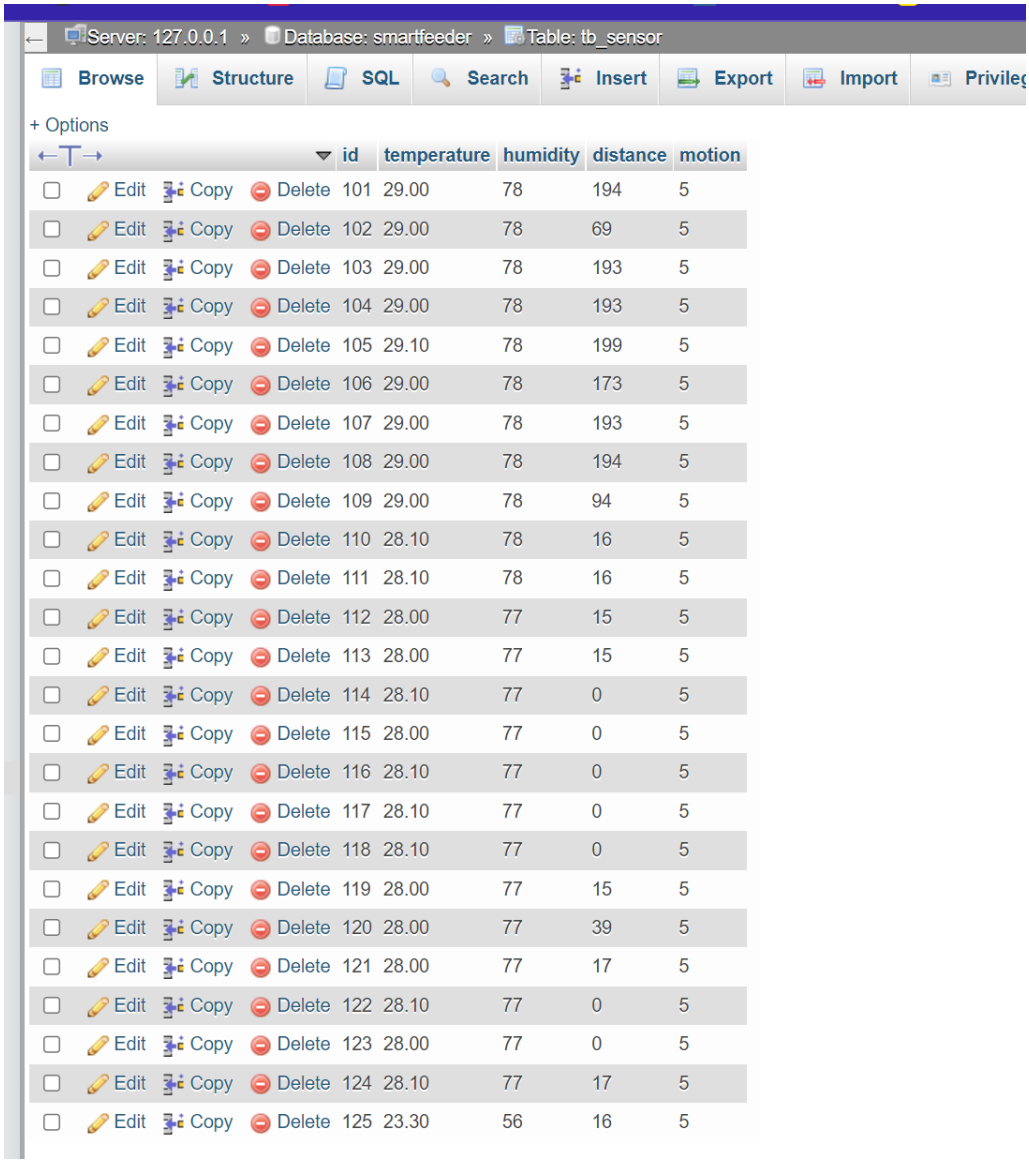
Figure 4.3.0: Result dashboard

This is my dashboard website. It serves to monitor the data with our own devices whether it's handheld device or PC. It has 4 cards to monitor the reading from sensor, temperature, humidity, distance and motion. DHT 11 shows the sensor reading of temperature and humidity. Ultrasonic display distance reading and servo motor display the reading of motion.

```
10:25:39.386 -> humidity : 56.00
10:25:39.431 -> distance : 16.1216.12
10:25:39.431 -> Distance (inch): 6.34
10:25:39.570 -> Success sent
10:25:40.593 -> temperature : 23.30
10:25:40.593 -> humidity : 56.00
10:25:40.640 -> distance : 16.1316.13
10:25:40.640 -> Distance (inch): 6.35
10:25:40.779 -> Success sent
10:25:41.806 -> temperature : 23.20
10:25:41.806 -> humidity : 55.00
10:25:41.853 -> distance : 16.1316.13
10:25:41.853 -> Distance (inch): 6.35
10:25:41.900 -> Success sent
```

Figure 4.3.1: Result serial monitor

This is the result of data I got from Serial Monitor in the Arduino IDE, as you can see, the data I got from this serial monitor is temperature, humidity, distance, and motion. It also showed that all the data was successfully sent to the database. This data is taken from my Final Year Project during Final Testing and verification for the presentation



	id	temperature	humidity	distance	motion
Edit Copy Delete	101	29.00	78	194	5
Edit Copy Delete	102	29.00	78	69	5
Edit Copy Delete	103	29.00	78	193	5
Edit Copy Delete	104	29.00	78	193	5
Edit Copy Delete	105	29.10	78	199	5
Edit Copy Delete	106	29.00	78	173	5
Edit Copy Delete	107	29.00	78	193	5
Edit Copy Delete	108	29.00	78	194	5
Edit Copy Delete	109	29.00	78	94	5
Edit Copy Delete	110	28.10	78	16	5
Edit Copy Delete	111	28.10	78	16	5
Edit Copy Delete	112	28.00	77	15	5
Edit Copy Delete	113	28.00	77	15	5
Edit Copy Delete	114	28.10	77	0	5
Edit Copy Delete	115	28.00	77	0	5
Edit Copy Delete	116	28.10	77	0	5
Edit Copy Delete	117	28.10	77	0	5
Edit Copy Delete	118	28.10	77	0	5
Edit Copy Delete	119	28.00	77	15	5
Edit Copy Delete	120	28.00	77	39	5
Edit Copy Delete	121	28.00	77	17	5
Edit Copy Delete	122	28.10	77	0	5
Edit Copy Delete	123	28.00	77	0	5
Edit Copy Delete	124	28.10	77	17	5
Edit Copy Delete	125	23.30	56	16	5

Figure 4.3.2: Result Database

This is the database for the sensors I got from my dashboard, as you can see here it have all the data I got earlier in serial monitor. Temperature, humidity, distance and motion. This data is stored in one private server so that no one can steal the data from the database.

4.4 Analysis

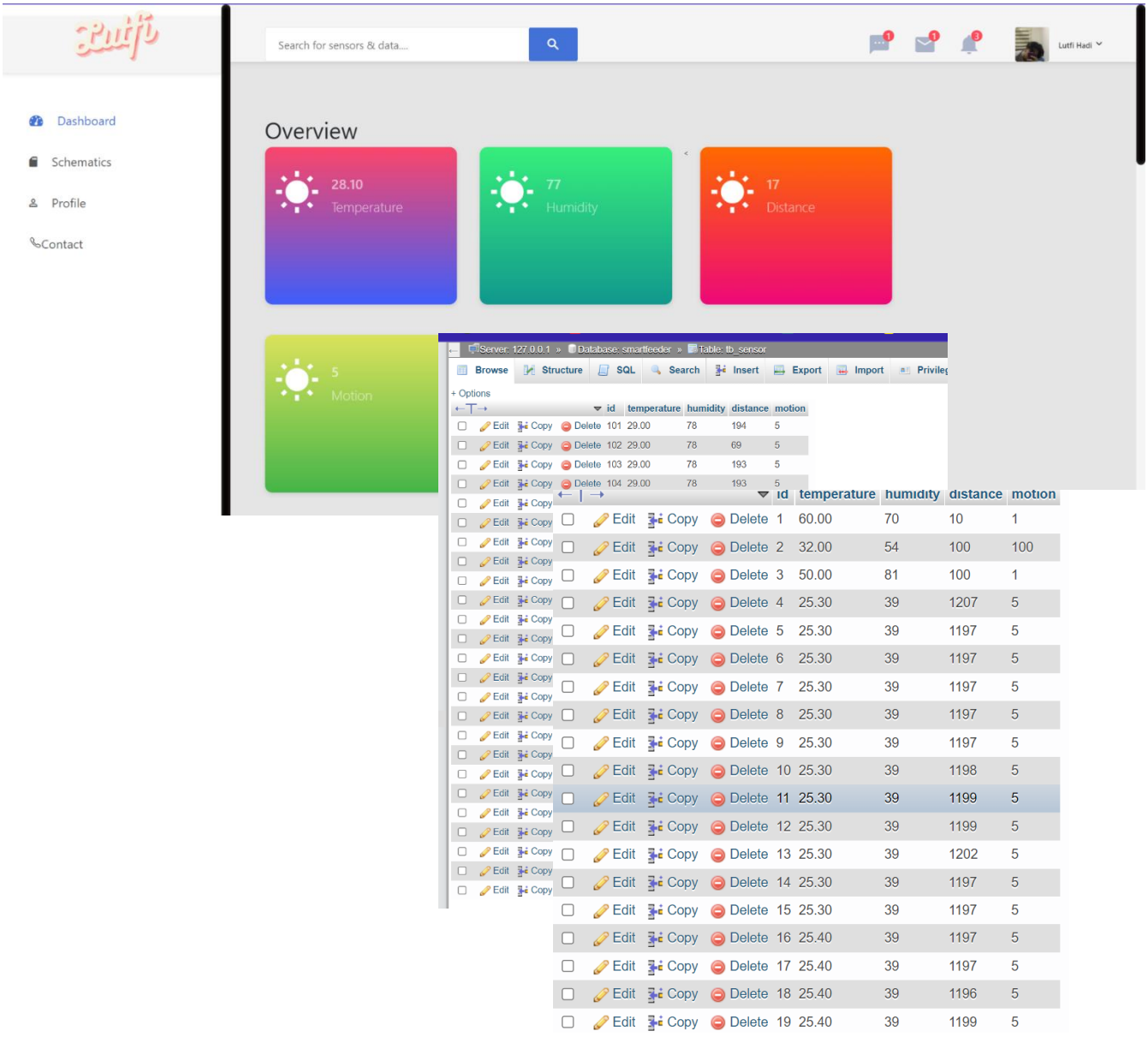


Figure 4.4.0: Temperature and Humidity Analysis

This is the analysis I can explained, as you can see in the database. The difference in temperature is because this Automatic feeder is placed in the place where it has air conditioner. The place like office or classroom with AC will decrease the temperature data and increase the humidity data.

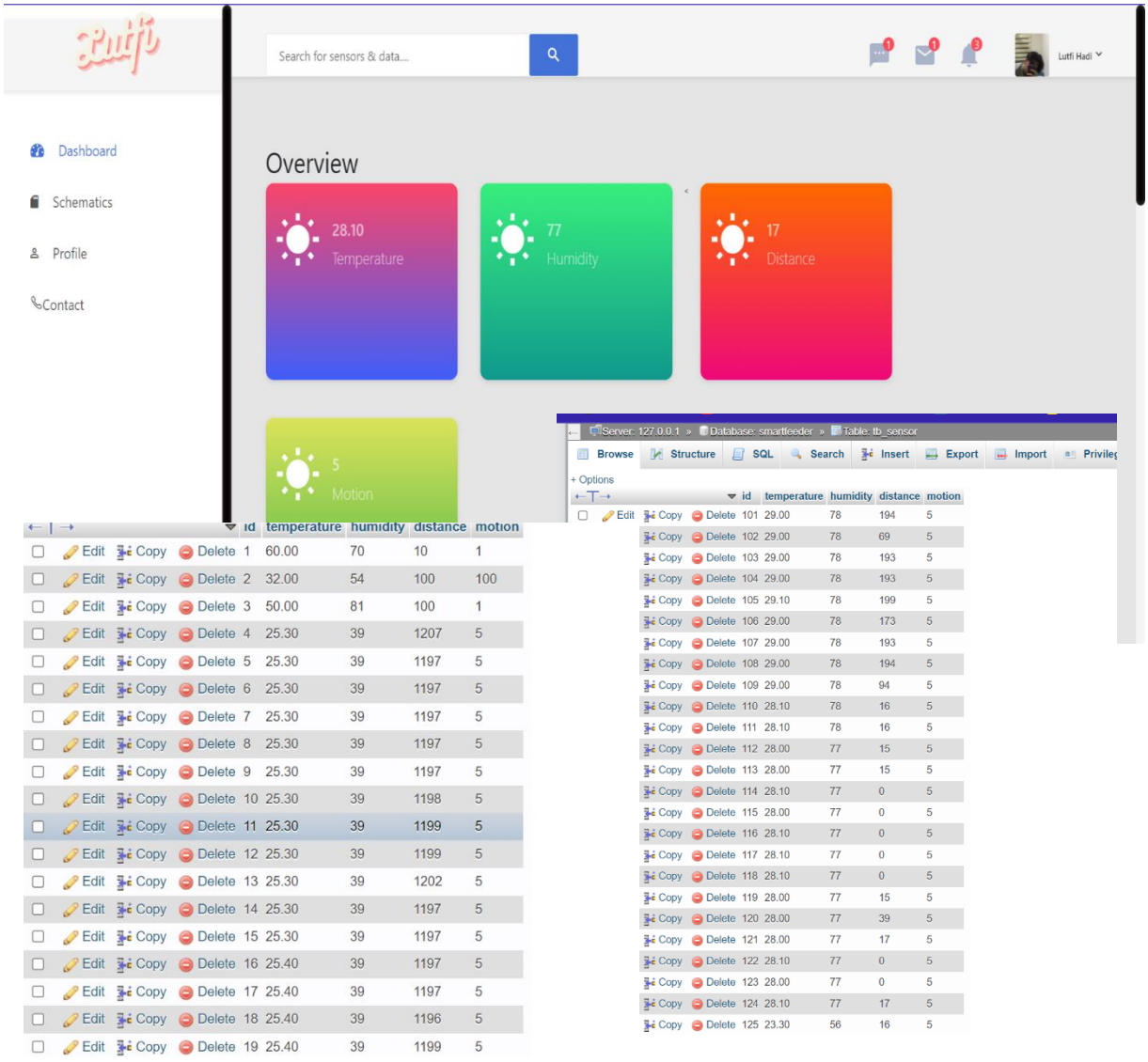


Figure 4.4.1: Food level analysis

This is the analysis about distance , in the database the distance is continuously decreasing . it's because the food level in the container is continuously decreasing. Hence the data of the distance will always decreasing over time.

4.5 Problem & Solution

Figure 4.5.0: Servo Attachment

Problem: Difficulty to attach the cardboard to the servo

Solution: I used araldite glue to attach the cardboard. It's really strong glue it can bonds anything to anything with ease. It also dry quick for faster bonds.

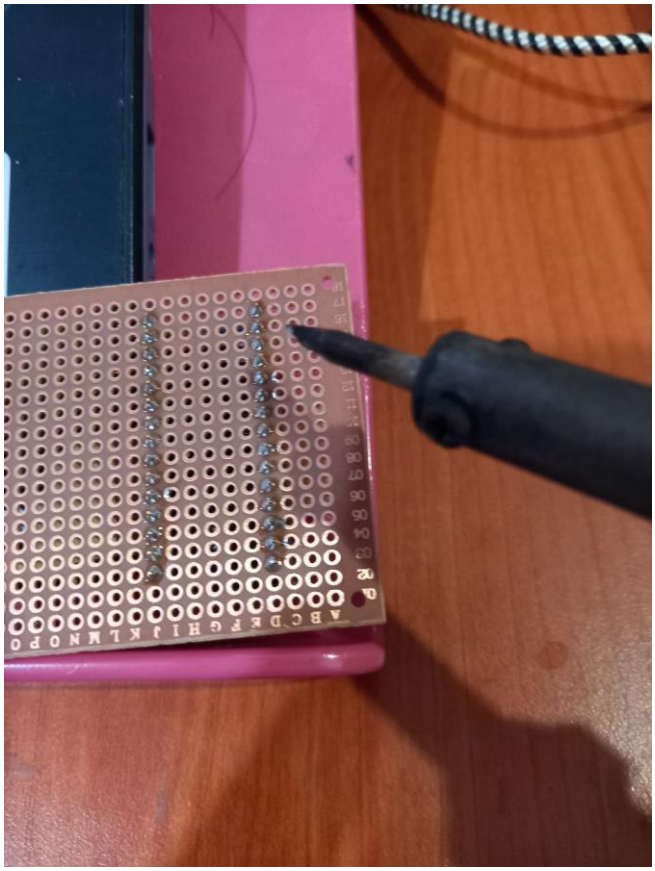


Figure 4.5.1: Stripboard and Solder iron

Problem: Having trouble to solder using only solder iron and solder wire.

Solution: By using Solder Flux you can solder the solder iron without any difficulty, by using flux it also can save your time.

CHAPTER 5
DISCUSSION, CONCLUSION & RECOMMENDATIONS

5.1 Discussion

In planning an activity and work related to the project, a discussion between supervisor and lecturer is a must to ensure the effectiveness and the activity goes smoothly as I planned. Each week there will be a meet up with my supervisor to discuss about my latest progress on project development and report correction. Furthermore, all the problems encountered by me some of it will be review by my supervisor, Encik Noran Zahrine. The problems are including sensor and actuator malfunction and design problem, the reason this problem was deliver to my supervisor is because I want to get suggestions of how to solve the problem perfectly.

With this, the issues and development of this project can be shared. All doubts and problem regarding the project are discussed each week until I reached the conclusion.

5.2 Conclusion

The interaction between pets, object and physical devices is going to attract and trigger the attention. Many studies have attempted to provide natural and intuitive approach to request service. The new technology combining pet dispenser with IoT Technology offers exciting ways for future development for pet feeder. The result is not only present the key improvement of the Automatic Pet feeder Monitoring system involved in the IoT technology but also can store the data that captured it's also meet the demand of pet owners

The basic vision behind the IoT.it may have a new way of operational method It may have a new way of connecting device using Automatic feeder. The monitoring of the quality and quantity of food is also important to ensure the IoT development is implemented in this project. Besides it is less useful if we are monitoring without storing the data using cloud. In the future we will centralize on the study of IoT Gateway and long detection motion of pets

5.3 Recommendation

All IoT project always have a room to improve by time, the technology is still improving until today just like my project, smart pet feeder it could have a lot of improvement but a lot of factors prevent me to improve my project to the limit many things could have been done to the project.

My IoT Smart feeder have a lot of things to improve, for examples I could have used Blynk apps to control the timer for my pet so that the food wastage can be avoided. For this Project the timer is more suitable to trigger the servo rather than sensor[7]. In addition, I could have added functionalities like a camera to detect absence of cat, it's also good for the owner to check whether their pets eats the food given of not. Finally, I think it's also recommended to put RFID tag on pet's neck [8]so whenever their pet is hungry, the pet will go to the Automatic Food dispenser and RFID scanner will trigger the servo to serve the food.

REFERENCES

[1]D. Plus, “Smart Automatic Pet Feeder With Arduino Uno,” *Device Plus*, 20-May-2022. [Website].

[2]lady Ada, “PIR Motion Sensor,” *Adafruit Learning System*, 28-Jan-2014.[Website].

[3] Apoorve. “What Is a Servo Motor? - Understanding the Basics of Servo Motor Working.” *Servo Motor Basics, Working Principle & Theory*, 1 Aug. 2015, [Website]

[4]lady Ada, “DHT11, DHT22 And AM2302 Sensors,” *Adafruit Learning System*, 29-Jul-2012. [Website].

[5]Keyence, “What Is An Ultrasonic Sensor? | Sensor Basics: Introductory Guide To Sensors | KEYENCE,” *What Is An Ultrasonic Sensor? 08-February-2020.* [Website].

[6]S. Santos, “ESP32 Vs ESP8266 - Pros And Cons - Maker Advisor,” *Maker Advisor*, 06-Sep-2021. [Website].

[7]Oliverb, “Iot pet feeder using the BLYNK Mobile App & an ESP8266 module”. 29-Sep-2019. [Website].

[8]Kumar Raushan, *Iot pet feeder using the Blynk mobile app and an ESP32 Mobile App and an ESP32*. IOT Pet Feeder Using the Blynk Mobile App and an ESP32. 05-Oct-2019. [Website]

[9]Sangvanloy, Tannop, and Kingkarn Sookhanaphibarn. “Automatic Pet Food Dispenser by Using Internet of Things (IOT).” *IEEE Xplore*, IEEE, 30 Apr. 2020. [Website]

[10]Own, Chung-Ming, et al. “The Study and Application of the IOT in Pet Systems.” *SCIRP Open Access*, Scientific Research Publishing, 29 Jan. 2013. [Website]

[11] Jost, Danny. “What Is an Ultrasonic Sensor?” *Fierce Electronics*, 7 Oct. 2019, [Website]

[12] javatpoint. “Xampp Tutorial - Javatpoint.” *Www.javatpoint.com*, 6 Sept. 2015,[Website]

Terminology

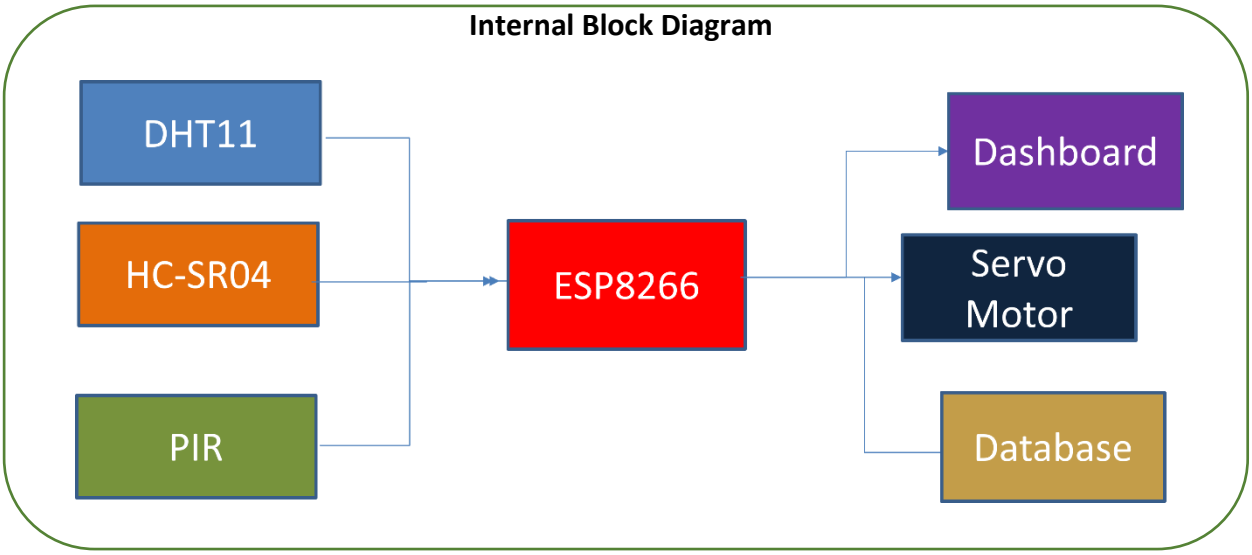
Node MCU	Compared to the ESP8266, the ESP32 is substantially more powerful, has more GPIOs with different functionality, quicker Wi-Fi, and supports Bluetooth. However, because the ESP32 is more complicated than the ESP8266[6], many people believe that it is more challenging to work with. Contrarily, in our perspective, programming the ESP32 is just as simple as programming the ESP8266, particularly if you plan to use MicroPython or the "Arduino language."
PIR motion sensor	PIRs are essentially constructed of a pyroelectric sensor, which can detect amounts of infrared radiation. You can see this sensor below as the round metal container with the rectangular crystal in the centre. Every object produces some low-level radiation, and the more radiation is emitted by an object the hotter it is. A motion detector actually has two sides to its sensor. This is because we want to detect motion (change) rather than an average of IR levels[12]. The wiring of the two sections makes them cancel each other out. The output will swing high or low depending on whether one part detects more or less IR radiation than the other.
Servo Motor	A servo motor is a kind of motor that has extremely precise rotational capabilities. This type of motor typically has a control circuit that gives feedback on the motor shaft's present location. This feedback enables the servo motors to rotate very precisely. A servo motor is used to rotate an object at predetermined angles or distances. It consists of a straightforward motor that drives a servo mechanism[3]. A motor is referred to as a DC servo motor if it is powered by a DC power source, and an AC servo motor if it is driven by an AC power source. We will solely talk about the operation of the DC servo motor in this lesson.
DHT 11	The DHT11 is a basic, ultra low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and spits out a digital signal on the data pin (no analog input pins needed). It's fairly simple to use but requires

	careful timing to grab data[4]. The only real downside of this sensor is you can only get new data from it once every 2 seconds, so when using our library, sensor readings can be up to 2 seconds old.
Ultrasonic Sensor (HC-SR04)	An ultrasonic sensor is a piece of technology that uses ultrasonic sound waves to measure a target object's distance and then turns the sound that is reflected back into an electrical signal[11]. The speed of audible sound is greater than the speed of ultrasonic waves (i.e. the sound that humans can hear). The transmitter (which generates sound using piezoelectric crystals) and the receiver are the two major parts of an ultrasonic sensor (which encounters the sound after it has travelled to and from the target).
XAMPP	One of the popular cross-platform web servers is XAMPP, which enables programmers to construct and test their applications on a local web server. It was created by the Apache Friends, and users can edit or change the native source code. It consists of the MariaDB database, the Apache HTTP Server, and interpreters for various programming languages like PHP and Perl. It is supported by various systems[12], including the x64 package of macOS and Linux and the IA-32 package of Windows, and it is accessible in 11 different languages.
Visual Studio Code	The source code editor in Visual Studio Code is blazingly fast and ideal for everyday usage. VS Code supports hundreds of languages and offers features like syntax highlighting, bracket matching, auto-indentation, box selection, snippets, and more to help you get started quickly. You can easily browse your code thanks to intuitive keyboard shortcuts, simple customisation, and keyboard shortcut mappings supplied by the community.

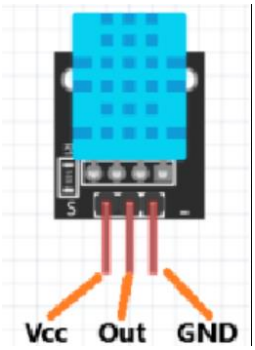
Table Terminology: Terminology

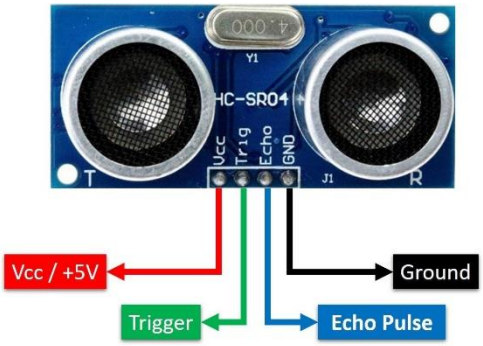
APPENDICES

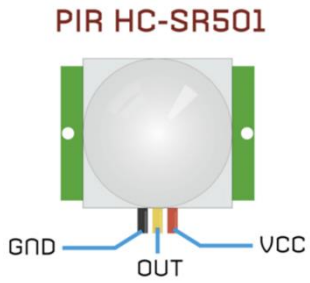
Internal Block Diagram



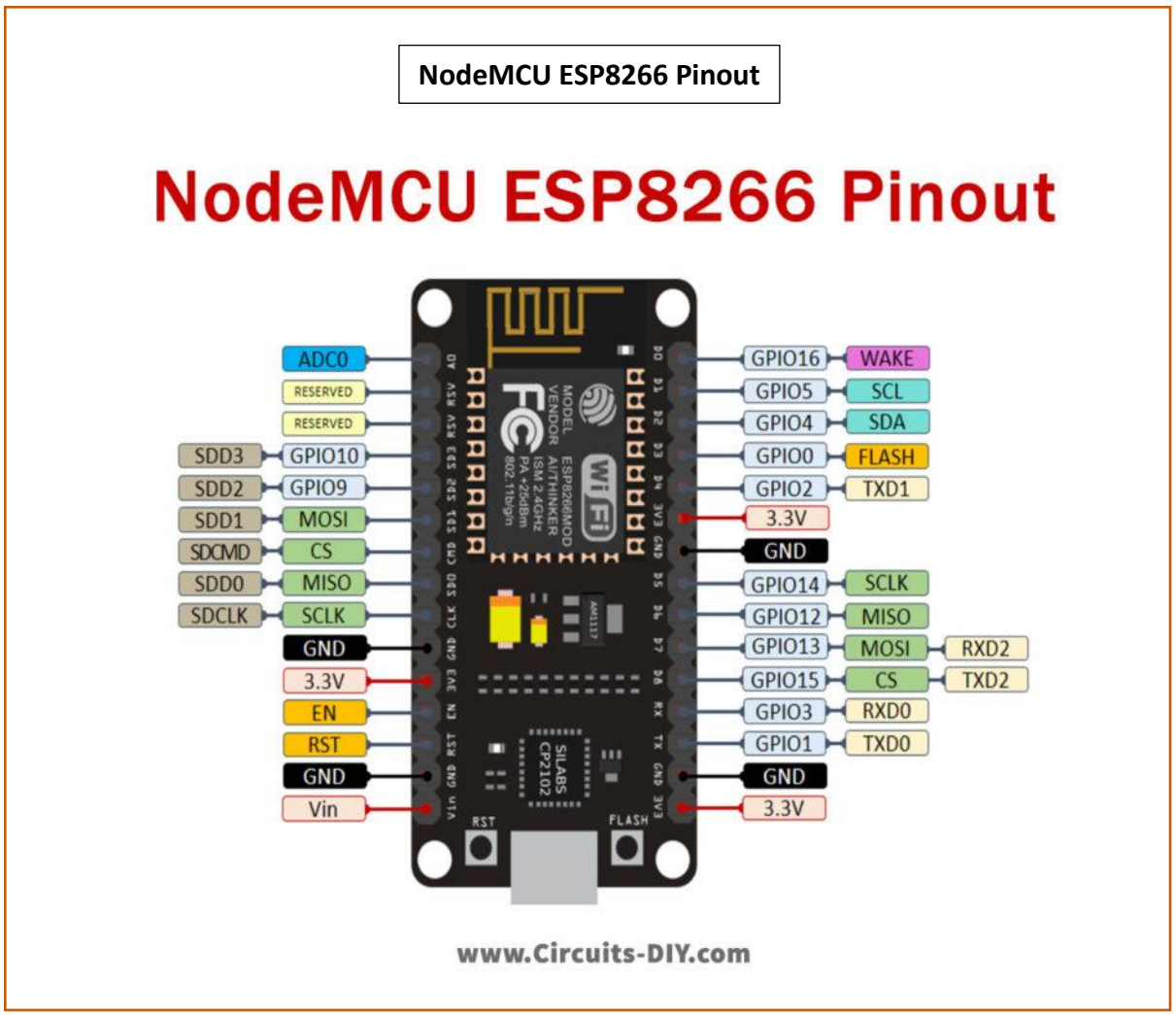
Pin Input & Output







NodeMCU ESP8266 Pinout



Final Year Project Video Demonstration

SCAN ME

Coding Arduino

```
#include <ESP8266HTTPClient.h>
#include <ESP8266WiFi.h>
#include <Servo.h>
Servo myservo;
int pos = 0;
int calibrationTime = 30;
long unsigned int lowIn;
long unsigned int pause = 5000;

boolean lockLow = true;
boolean takeLowTime;

int motion = 5;            //digital pin connected to the PIR's
output
int pirPos = 13;           //connects to the PIR's 5V pin


//include dht library

#include "DHT.h"

#define DHTPIN 2
#define DHTTYPE DHT11  // DHT 11

DHT dht(DHTPIN, DHTTYPE);

#define SOUND_VELOCITY 0.034
#define CM_TO_INCH 0.393701

long duration;
float distance;
float distanceInch;

const int trigPin = 12;
const int echoPin = 14;

//wifi credential

const char* ssid = "Lutfi";
const char* password ="Lutfilelouch00";
const char* host = "172.20.10.2";


void setup() {
 pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
 Serial.begin(9600);
 dht.begin();

// wifi
WiFi.begin(ssid, password);
Serial.println("Connecting....");
while(WiFi.status() !=WL_CONNECTED)
{
  Serial.print(".");
  delay(500);

  }
```

```
//success

Serial.println("Connected");



{
 myservo.attach(15);    //attaches servo to pin 8
  Serial.begin(9600);    //begins serial communication
  pinMode(motion, INPUT);
  pinMode(pirPos, OUTPUT);
  digitalWrite(pirPos, HIGH);


 //give the sensor time to calibrate
  Serial.println("calibrating sensor ");
  for(int i = 0; i < calibrationTime; i++){
    Serial.print(calibrationTime - i);
    Serial.print("-");
    delay(1000);

  }
}

Serial.println();
  Serial.println("done");

 while (digitalRead(motion) == HIGH) {
    delay(500);
    Serial.print(".");
  }
  Serial.print("SENSOR ACTIVE");
}

void loop() {

 //dht
float temperature = dht.readTemperature();
float humidity = dht.readHumidity();


Serial.println("temperature : " + String(temperature) );
Serial.println("humidity : " + String(humidity) );

{


  if(digitalRead(motion) == HIGH){  //if the PIR output is HIGH, turn
servo

        for(pos = 0; pos < 180; pos += 1)  //goes from 0 to 180
degrees
        {
           myservo.write(pos);                  //tells servo to go
to position in variable "pos"
      delay(5);
        }
        for(pos = 180; pos>=1; pos-=1)     //goes from 180 to 0
degrees
```

```
{
   myservo.write(pos);
    delay(5);
}

 if(lockLow){
      //makes sure we wait for a transition to LOW before further
output is made
      lockLow = false;
      Serial.println("---");
      Serial.print("motion detected at" + String(motion) );
      Serial.print(millis()/1000);
      Serial.println(" sec");
      delay(50);
 }
  takeLowTime = true;

}

 if(takeLowTime){
      lowIn = millis();             //save the time of the transition
from HIGH to LOW
      takeLowTime = false;    //make sure this is only done at the
start of a LOW phase
    }

  //if the sensor is low for more than the given pause,
    //we can assume the motion has stopped
    if(!lockLow && millis() - lowIn > pause){
      //makes sure this block of code is only executed again after
      //a new motion sequence has been detected
      lockLow = true;
      Serial.print("motion ended at " + String(motion)); //output
      Serial.print((millis() - pause)/1000);
      Serial.println(" sec");
      delay(50);


     // Wait a few seconds between measurements.
  delay(2000);

}

{
  // Clears the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Reads the echoPin, returns the sound wave travel time in
microseconds
  duration = pulseIn(echoPin, HIGH);

  // Calculate the distance
  distance = duration * SOUND_VELOCITY/2;

  // Convert to inches
```

```
  distanceInch = distance * CM_TO_INCH;

  // Prints the distance on the Serial Monitor
  Serial.print("distance : " + String(distance) );
  Serial.println(distance);
  Serial.print("Distance (inch): ");
  Serial.println(distanceInch);

}

//send data to server
WiFiClient client ;

//port 80
const int httpPort = 80;
if(!client.connect(host, httpPort))

{
  Serial.println("Connection Failed");
  return;

  }

  //if connected
  //send data to database

  String Link;
  HTTPClient http;

  Link= "http://" + String(host) +
"/iotsmartfeeder/senddata.php?temperature="+ String(temperature) +
"&humidity=" + String(humidity) + "&distance=" + String(distance)  +
"&motion=" + String(motion);


  //execution link

  http.begin(Link);
  http.GET();

  //read response after finished uploading

  String response = http.getString();
  Serial.println(response);
  http.end();


delay(1000);
}
}
```