AUTOMATED SUPERWORM SEGREGATION SYSTEM FOR MULTI-PLASTIC BIODEGRADATION USING ESP32 WITH COMPUTER VISION FOR REAL-TIME MONITORING VIA MOBILE APPLICATION

A Project Study Presented to the Faculty of

Electronics Engineering Department College of Engineering

Technological University of the Philippines

In Partial Fulfillment of the Course Requirements for the Degree of

Bachelor of Science in Electronics Engineering

Project Study Proponents:

Angeles, John Cedric M.

Barretto, Carlo E.

Gamad, Serafin Raffy C. III

Manlutac, Leandro C.

Mayuno, Aira Mari W.

Project Study Adviser:

Engr. Jay Fel C. Quijano

August 2023

APPROVAL SHEET

This project study is entitled **"AUTOMATED SUPERWORM SEGREGATION SYSTEM FOR MULTI-PLASTIC BIODEGRADATION USING ESP32 WITH COMPUTER VISION FOR REAL-TIME MONITORING VIA MOBILE APPLICATION"** has been prepared and submitted by the following proponents:

Angeles, John Cedric M.

Barretto, Carlo E.

Gamad, Serafin Raffy C. III

Manlutac, Leandro C.

Mayuno, Aira Mari W.

In partial fulfillment of the requirements for the degree of **Bachelor of Science in Electronics Engineering** is hereby recommended for approval.

_____

ENGR. JAY FEL C. QUIJANO
Project Study Adviser

_____ _____

ENGR. JOHN PETER M. RAMOS
Panel Chair

ENGR. CHERRY G. PASCION
Panel Member

_____ _____

ENGR. EDGAR A. GALIDO
Panel Member

ENGR. GILFRED ALLEN M. MADRIGAL
Panel Member

_____ _____

ENGR. ROMEO L. JORDA JR.
Panel Member

ENGR. LEJAN ALFRED ENRIQUEZ
Panel Member

Accepted and approved in partial fulfillment of the course requirements for the degree of **Bachelor of Science in Electronics Engineering**

_____ _____

ENGR. TIMOTHY M. AMADO
Head, ECE Department

ENGR. NILO M. ARAGO
Dean, College of Engineering

ABSTRACT

Almost 80% of the plastics in the period of 1950-2017 became plastic waste, and ended up in landfills or oceans as per the United Nations Environment Programme. In this research, the researchers have used the capability of Superworms, or Zophobas morio, to safely biodegrade different types of plastics and produce Frass that can also be used for fertilizers, for the healthy cycle in the environment, tackling the SDG 15, Climate Action, The researchers developed a machine that automated their life cycles, distributed 4 types of plastic (Polystyrene (PS) , Expanded Polystyrene (EPS), Polyethylene (PE), and Polyurethane (PU)) that were fed to the Superworms, and provided ideal living conditions. These processes were done through different mechanisms that were controlled by an ESP32 and a mobile application for monitoring. The effectiveness of the reproduction process was based on the detection and counter feature of computer vision, it was able to count the worms in the container and had a confidence rate of 60% and counted with 5% margin of error, representing that the breeding aspect of the system is effective. In the data gathered daily, the plastic consumption of the 4000 Superworms were 5.72g, 5.72g, 2.28g, 0.92g, for PS, EPS, PU, and PE, respectively. Showing that plastic biodegradation is most effective with the PS plastics. The prototype showed desirable results as it provided an ideal living condition for the organisms, as it maintained a temperature range of 28-32 C.

ACKNOWLEDGEMENT

The proponents of this research would like to give immense gratitude towards these individuals who have been giving their full support to the proponents and, in extending their unwavering support, trust, and guidance for the successful completion of this research.

To God Almighty, in providing strength and wisdom to continue to persevere, and overcome the hurdles that the proponents experienced all throughout the research.

To our research adviser, Engr. Jay Fel C. Quijano, for his wisdom, patience, and never-ending encouragement to improve each day. To his enthusiasm and responsible leadership to the group, words can't express how fortunate it is to have him as a research adviser.

To the Dean of the College of Engineering in the Technological University of the Philippines - Manila, who is also the Capstone professor, Engr. Nilo M. Arago, for his dedication and guidance in partaking his knowledge in the research and capstone field.

To all the faculty of the Electronics Engineering department, for all of the lessons and principles that they taught that enabled the researchers to plan and finish the project.

To the Barretto family for always accepting us in their humble home and giving us the place for us to meet and finish our project.

Last but not the least, to our family and friends, who have been always supporting us through every aspect, may it be financially, emotionally, and physically. Your limitless love and dedication gave us the strength to bring forth success in this project.

TEAM SUPERB

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

The Problem and Its Setting

This chapter presented the introduction, background of the study, research gap, research objectives, significance of the study, and the definition of terms.

1.1 Introduction

In today's world, the rapidly growing population, extensive urbanization, and rising living standards, along with the widespread use of plastics, Styrofoam, and other similar materials, have made non-biodegradable wastes a chronic global concern. Pollution is one of the most prevalent problems, whether it be air pollution, water pollution, noise pollution, etc. More and more people are becoming aware of how detrimental the adverse effects of these pollutants can be. Thus, non-biodegradable resources are substances that cannot be dissolved by natural organisms and hence serve as a source of pollution or contamination. Nowadays, about 85 percent of all materials produced around the globe end up in landfills or incinerators [1]. Plastics account for a significant portion of waste materials, and recent estimates indicate that more than 100 million metric tons of polymers are generated globally each year [1].

An estimated 12 million tonnes of plastic are dumped into our seas, with Asia having more than 80% of accidents in marine leakage in the world. And unmanaged of

almost 1 billion metric tons of unmanaged plastic flowing into our seas each year. From the statistics of every country, the Philippines placed third as the largest contributor of unmanaged plastic. This issue increased public consciousness of plastic waste and management, bringing plastic pollution to the consciousness of the Philippine consumers [2]. Numerous plastics and Styrofoam containers made of polystyrene are non-biodegradable. Polystyrene, a phenomenally popular form of plastic, can resist very harsh circumstances while refusing to degrade, and environmentalists have pushed authorities to ban or severely limit its manufacture since it tends to appear in the most inconvenient places. Recent research throws some insight into how long it may last, demonstrating that even under the best of conditions, a single piece of polystyrene trash can last for decades [3]. Styrofoam is notoriously difficult to recycle, according to one estimate, Styrofoam might take up to 30% of the area in some landfills. Some estimates place the lifetime of Styrofoam at roughly 500 years [4].

Other types of plastics such as Polyethylene (PE), Polyurethane (PU), Polyphenylene sulfide (PPS), and Polypropylene (PP) had contributed to the growing percentage of plastic pollution. Polyethylene or polythene (PE), IUPAC name polyethylene or poly (methylene), is the most common plastic. Over 100 million tons of PE resins were manufactured yearly in 2017, accounting for 34% of the overall plastics industry [5]. Many kinds of PE are known, with most having the chemical formula (C2H4) n, such as soft drinks and water bottles, plastic bags, garbage bags, plastic cups, grocery bags, etc. These concerns brought the researchers to think of a way of solving the plastics and Styrofoam problem. A study that belongs to the Advances in Biocompatible and Biodegradable Polymers shows that there are organisms, such as the larvae of *Tenebrio Molitor* (yellow

mealworm) and *Zophobas atratus* (superworm), that possess a biodegradation ability for polystyrene (PS) [6].

Superworms may offer a key to solving the massive plastics and Styrofoam products problem. Superworms are larvae of the Zophobas Morio beetle. These are one of the largest insects with a body size of around 6-8 cm long. They have 4 distinct life stages: larvae, eggs, pupa (cocoon), and beetle. These organisms can spend a significant portion of their existence as larvae. They will not pupate if they come into touch with too many other larvae, consequently, keeping the superworms apart helps them to mature into beetles. The eggs of Zophobas Morio are oval-shaped, white in color, and with a length of 1.7 mm and a width of 0.7 mm. During her lifetime, each female may deposit a large number of eggs (up to 2,200). Larvae or the Superworms have a dark brown lining with a body of golden color and a segmented body.The pupae are mainly dormant, but when touched, they can circularly twist their abdominal segments or display other physical reactions. All of these responses, which are induced by the stimulation of distinct types of mechanoreceptive sensilla on the pupal body surface, are thought to represent an efficient pupal defensive system against predator assaults and larval cannibalistic activities. Lastly, adults are large (38-mm to 57-mm body length) with elongated bodies and filiform antennae [7].

Tiny mealworms may hold part of the solution to the giant plastics problem. Not only are they able to consume various forms of plastic, but they can also eat polystyrene containing a common toxic chemical additive and still be safely used as protein-rich feedstock for other animals. Meanwhile, "superworm" (Zophobas Morio larvae), a species with a lifetime of 6 months, can eat Styrofoam much greater than mealworms. In a study

from Science of the Total Environment, over 28 days, superworms could survive on Styrofoam alone as well as on a conventional diet (bran). The average Styrofoam consumption rate for each superworm was determined to be 0.58 mg/d, which was four times that of the mealworm [8].

1.2 Background of the Study

Concerns about the rising amount of plastic pollution have prompted experts all around the world to devise a method for degrading these compounds, which are said to take decades to break down. To counteract this issue, research on plastic biodegradation by bacteria and fungus has steadily increased. Microbial enrichment and isolation investigations have shown that numerous bacterial isolates are capable of digesting plastics, however, rates of breakdown vary and are often modest [9].

Recent research has shown that mealworms (Tenebrio Molitor larvae) from diverse sources throughout the world rapidly eat and biodegrade polystyrene (PS) to CO2 and lower molecular weight chemicals within their gut [9]. On the other hand, another type of darkling beetle can work in the same way as mealworms. A 2021 study shows that Superworms can survive and thrive on Styrofoam alone, the average Styrofoam consumption rate for each superworm was determined to be 0.58 mg/d, which was four times that of the mealworm [8].

In an article that belongs to the Advances in Biocompatible and Biodegradable Polymers, during the 30-day trial, the superworm had the largest PS consumption capacity and the highest survival rate. They could all damage PS to varying degrees. The superworm had the greatest capacity to break down PS into low-molecular-weight compounds,

whereas yellow mealworms strongly depolymerized PS by breaking the benzene ring [6]. In another study conducted in 2020, plastic biodegradation by fungal and bacterial strains has been highlighted as a viable approach for removing plastic trash without causing secondary pollution. A Pseudomonas aeruginosa strain obtained from the gut of a superworm was shown to be capable of degrading polystyrene (PS) and polyphenylene sulfide (PPS). They show how P. aeruginosa has outstanding biodegradative capacity by successfully depolymerizing four distinct kinds of plastics: PS, PPS, polyethylene (PE), and polypropylene (PP) [10].

Polyethylene (PE), is the most common plastic in the world, a study from the Circular Agricultural Systems shows how different types of insects contribute to degrading various types of plastics. Galleria mellonella moth larvae have the highest rates (mean 5.9 milligrams per gram live weight per day, standard error 1.5, N = 5). 4.3 mg/g/d was observed in a single investigation of Zophobas atratus beetle larvae. The mean rate of Tenebrio Molitor beetle larvae was 1.7 mg/g/d (standard error 0.4, N = 7) [11].

The automation of insect rearing, collection, and handling is a critical step in making insect mass production more convenient, cost-effective, and competitive. An automated mass rearing system may encourage mass organism growth from egg hatching to full adulthood or certain stages in between, such as the larvae rearing process, with little to no human interaction. By automating the growing and transportation of these species, deaths and developmental problems may be minimized. The discovery that superworms can biodegrade different types of plastics is now a step toward converting non-biodegradable pollutants into decomposable waste. This study is successful because it employed automated technology to create a system that can continually biodegrade plastics

and/or Styrofoam while also providing a rearing mechanism for the darkling beetle's life cycle

1.3 Research Gap

Styrofoam and other plastic wastes are one of the hardest materials to dispose of, thus making a fully automated multi-plastic biodegradation system is needed. An ideal biodegradation system would be a fully automated plastic distributor, wherein these plastics are automatically distributed based on the specific amount of each plastic in each container. Another is a real-time monitoring system feature to carefully monitor and check the state of the ecosystem. The integration of these features is one of the most important solutions that the researchers provided, in which the previous researches did not possess. One of the main challenges the researchers faced with these innovations and improvements is the building and assembly of the food distributor mechanisms, as well as the development of the automation for the breeding system with an accompanied computer vision feature. Thus, a fully automatic multi-plastic biodegradation using superworms is proposed.

1.4 Research Objectives

To develop a fully-automated Superworm segregation system for multi-plastic biodegradation using various sensors and mechanical devices, as a solution to plastic pollution.

In particular, it aims to:

1. To develop an ESP32 based system that will automate the breeding and segregation of Superworms for multi-plastic biodegradation using humidity and temperature sensor, webcam, weight sensor, servo motor, vibration motor, and linear actuator.
2. To develop a mobile application that will notify the user, control parameters, and monitor the system with computer vision that will aid in ensuring the quality of the state of the organisms and the system.
3. To determine the system's efficiency in terms of plastic biodegradation, and its ability in automating the segregation of life stages of the Superworms.
4. To establish a business to consumer model in maximizing the marketability of the superworms and its byproduct, frass.
5. Test and Evaluate the prototype using the ISO Standard 25063.

1.5 Significance of the Study

The researchers see the potential to create a completely automated multi-plastic biodegradation system for the Zophobas Morio.

This research should have a significant impact on the following areas: The Material Recovery Facilities (MRF) of Local Government Units are the principal end-users of this

technology. This supplied a solution to one of the community's most problematic plastics and/or white pollution concerns because this study required less time and energy to degrade plastic and Styrofoam than other techniques. Furthermore, this became advantageous by developing a completely automated artificial habitat for superworms that does not need a lot of work. This study might help the government allocate funds for waste collection and disposal. Since this project focuses on the biodegradation of plastics and Styrofoam products, which are deemed non-biodegradable, the development of a system that can disintegrate the aforementioned substances can benefit the country economically.

Plastics such as polystyrene, polyethylene, etc. influence the ecosystem and environment. Biodegradation has long been a global concern due to such materials' non-biodegradability. When plastics and Styrofoam are burnt, it produces toxic gasses that endanger humans, animals, and the environment. This study exemplified the use of electronics engineering environmentally. The researchers proposed a solution that could help solve the growing percentage of resources that cannot be dissolved by natural organisms. This study addressed the goals for sustainable cities and communities, responsible consumption and production, climate action, life below water, and life on land, which are numbers 11, 12, 13, 14, and 15, respectively, provided in the Sustainable Development Goal (SDG) of United Nations.

This research falls under section IV: Industry, Energy, and Emerging Technology of Harmonized National Research and Development Agenda (HNRDA) provided by the Department of Science and Technology (DOST), specifically in the Delivery of Social Services, which focuses on the environment and pollution control. Furthermore, the study is involved in section III of HNRDA which is the Agriculture, Aquatic and Natural

Resources (AANR), specifically in the Crops R&D Agenda and Natural Resources and Environment R&D Agenda. Future researchers may use the concepts offered as a reference to evaluate the validity, enhance the project, or perform new studies.

1.6 Scope and Limitations

This study developed a fully automated device that will help in the biodegradation of different plastics, namely, Polystyrene(PS), Expanded Polystyrene (EPS), Polyethylene (PE), and Polyurethane (PU), using the plastic degrading ability of the superworms (Z. morio). This device is automated with the use of ESP32 for the segregation, collection module, breeding and hatching system, and the food dispensing. This device housed and segregated 4000 superworms by its life stages including its wastes or the frass.

In the food processing system, the device included various servo motors and load sensors in order to distribute the plastics with specific weight in specific containers for optimal placing and ingestion of the plastics. These mechanisms were also accompanied by 3D printed designs that aid in making the movements of these motors and sensors.

In the monitoring and notification system, a raspberry pi camera is installed in the incubation of larva container, and a webcam on the biodegradation container. The real-time monitoring system is displayed in the android application that the user will be using and includes a computer vision that will help in counting the organisms in the biodegradation container, this will help in ensuring that the system provides the ideal environment for the organisms.

The limitations of the project are the other parameters in ideal living conditions, other plastics that are not mentioned in the scope such as laminated plastics, because of the limited biodegradation capability of the organisms used, larger size, density and volume of

plastics due to the small size of the organisms and its consumption rate, and other worms that can degrade plastics as other organisms also have different ideal living conditions. The manual handling of the packing of the superworms and as well as the placement to the isolation compartments, due to its delicate and fragile condition. And only 4000 superworms are used as the controlled population in determining the plastic biodegradation of the study, and 6000 superworms is the population limit for the biodegradation container. The excess superworms will be transferred to other containers. Furthermore, the uncertain breeding of the beetles as this falls on the biological aspect. These are the limitations that the researchers have stumbled upon in developing this project.

1.7 Definition of Terms

For clarification purposes, the following were the definition of terms used by the researchers in the paper.

Beetle - an insect of an order distinguished by forewings typically modified into hard wing cases (elytra) that cover and protect the hind wings and abdomen.

Database - a structured set of data held in a computer, especially one that is accessible in various ways.

Expanded Polystyrene (EPS) - It is a type of white foam plastic made from solid polystyrene beads. Its primary applications include packaging, insulation, etc.

Frass - It is known as the waste products of insect larvae. This residual product is a helpful soil additive with beneficial levels of nitrogen, potassium, and phosphorus.

Larvae - the immature, wingless, and often wormlike feeding form that hatches from the egg of many insects, alters chiefly in size while passing through several

molts, and is finally transformed into a pupa or chrysalis from which the adult emerges.

Mealworms - It is the larva of darkling beetles (*Tenebrio molitor*), which infest grain products but are frequently raised as food for insectivorous animals, for use in laboratories, or as bait for fishing.

Polyethylene - It is the most commonly used plastic within the polyolefin family. It is utilized for plastic containers, plastic toys, plastic parts, bottles, bags, films, tubes, laminates, and more.

Polyphenylene sulfide (PPS) - It is a semi crystalline, high temperature thermoplastic polymer and is used in filter fabric for coal boilers, papermaking felts, electrical insulation, film capacitors, specialty membranes, gaskets, and packings.

Polypropylene (PP) - It is one of the thermoplastics that is commonly used worldwide. Uses for polypropylene include fibers and textiles as well as plastic packaging, plastic parts for machinery and equipment.

Polystyrene (PS) - It is a naturally transparent thermoplastic that can be found both in the forms of rigid foam and standard solid plastics. It is frequently used in the food service industry such as rigid trays and containers, disposable eating utensils, and foamed cups, plates, and bowls.

Polyurethane (PU) - One of the most widely used materials in home furnishings, including furniture, bedding, and carpet underlay, is polyurethane, typically in the form of flexible foam.

Pupa - is usually enclosed in a cocoon or protective covering, and undergoes internal changes by which larval structures are replaced by those typical of the imago

Superworms - are the larvae of darkling beetles, they serve as a food source for creatures including fish, reptiles, and birds.

Chapter 2

Review of Related Literature and Studies

This chapter presents the literature, journals, studies, and articles that have a correlation with the researchers' study.

2.1 Plastics

2.1.1. Polystyrene (PS)

Polystyrene (PS) is a polymer created from the monomer styrene, which is a liquid petroleum-based hydrocarbon. PS is normally a hard plastic but when presented in a higher temperature, it can be melted. PS is an aromatic polymer while styrene is an aromatic monomer [12].

Polystyrene is used in a variety of products, including appliances, automobile parts, electronics, food services, medications, and packaging, since it is affordable, resilient, and chemically inert.

2.1.1.1 Expanded Polystyrene (EPS)

Small polystyrene beads generated from styrene via a polymerization process make up EPS. Small spheres of polystyrene (from crude oil) with an expansion agent (e.g., pentane C6H12) that expand when heated with water vapor [13] are used to make expanded polystyrene (EPS).

In the construction industry and everywhere else, EPS is mostly utilized as a packing or insulating material. It has a low heat conductivity, making it a good insulating and transportable material [14]. EPS has a low density and a compressive strength of effectively zero.

2.1.2 Polyethylene (PE)

Polyethylene or PE is one of the most common plastics. 34% of the total plastic waste as of 2017 are PE plastics [5]. PE comes in a variety of forms, from hard to soft plastics, with high and low density.[15]. To identify the kind of PE plastics, density and branching are used. Every property of PE will have an effect on the mechanical properties of the plastic.

PE comes in a variety of forms, including ultra-high molecular weight polyethylene, ultra-low molecular weight polyethylene, high molecular weight polyethylene, high density polyethylene (HDPE), high density cross-linked polyethylene, cross-linked polyethylene, medium-density polyethylene, linear low-density polyethylene (LLDPE), low density polyethylene (LDPE), very low density polyethylene, and chlorinated polyethylene. The most important PE grades in terms of sold volumes are HDPE, LLDPE, and LDPE.

2.1.2.1 Market Size of Polyethylene

According to the market research report on Polyethylene Market size (2020) [16], the global polyethylene market was valued at $107.43

billion in 2019 and is expected to grow at a CAGR of 3.4 percent to $130.26 billion by 2027.

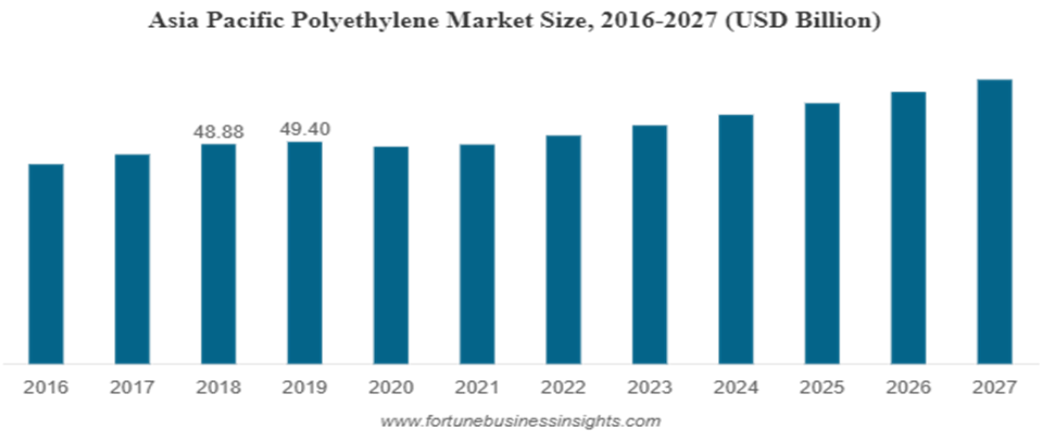


Figure 1. Market size of polyethylene

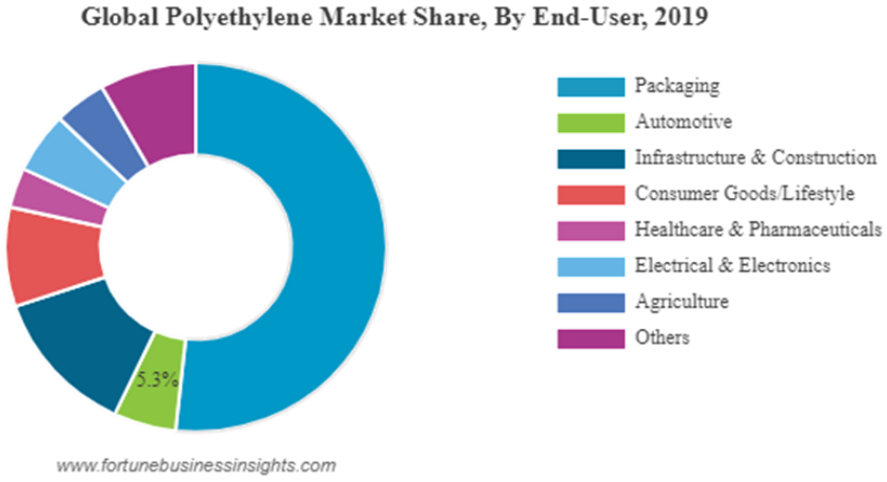


Figure 2. Global polyethylene market share

Figure 2 shows that more than half of the usage of global polyethylene market share is focused on packaging.

2.1.2.2 Low Density Polyethylene (LDPE)

LDPE is a type of commonly produced plastic made from ethylene and having a relatively low density, used chiefly for packaging, especially plastic bags. It is a soft, flexible, lightweight plastic material. LDPE is noted for its low temperature flexibility, toughness, and corrosion resistance. It is not suited for applications where stiffness, high temperature resistance and structural strength are required. It is often used for orthotics and prosthetics. LDPE has good chemical and impact resistance and is easy to fabricate and form [17].

2.1.2.3 High Density Polyethylene (HDPE)

High-density polyethylene is an ethylene linear addition polymer produced at temperatures and pressures like LLDPE, with only very slight branching. HDPE films are stiffer than LDPE films, despite remaining flexible, and have less transparency. HDPE has a higher working temperature than LDPE because of this, but it is also harder, stronger, and slightly heavier (but less ductile) [18]. It appears wax-like, lusterless, and opaque. Shampoo bottles, toys, chemical containers, and other common applications include HDPE [19].

2.1.2.4 Linear Low Density Polyethylene (LLDPE)

Linear low density polyethylene (LLDPE) is a copolymer of ethylene and a-olefin with linear sequences and a certain amount of short

chain branches responsible for its high crystallinity, endowing LLDPE with superior properties such as tensile strength, tear strength, and puncture resistance, [20] and has thus been widely used in the film industry. LLDPE resins are polyethylene (PE) plastic materials with densities in the 0.915–0.925 g/cm3 range.

2.1.3 Polyurethane (PU)

Polyurethane is a polymer with properties that exhibit elastomeric properties and are highly durable. PUs can be either thermoplastics or thermosets [21]. Because of its properties, PU has been used in wound dressings [22]. PUs is used to make foams, as well as coatings, adhesives, and varnishes.

2.1.4 Polyphenylene Sulfide (PPS)

Polyphenylene sulfide is a thermoplastic engineering material with good performance. PPS is tolerable and can be molded, extruded, or machined. In its pure solid state, it is an opaque white to light tan color. The maximum temperature for serving is 218 degrees Celsius. PPS does not dissolve in any solvent at temperatures below around 200 °C. PPS is one of the most important high-temperature polymers due to its multiple desired characteristics [23]. These qualities include resistance to heat, acids, and alkalies, as well as mildew, bleaches, aging, sunshine, and abrasion. It is color resistant and absorbs very little solvent. Composites made of poly (phenylene sulfide) are utilized in applications that require great strength and chemical resistance at high temperatures [23]. PPS (polyphenylene sulfide) is a

chemically resistant engineering thermoplastic with outstanding mechanical and thermal properties found in the electronics, automotive, aerospace, and chemical industries [24].

Table 1. Description of plastics

Name of Plastic	Description	Uses
Polyethylene terephthalate (PET)	Clear tough plastic, may be used as a fiber	Soft drink and water bottles, filling for sleeping bags and pillows, textile fibers
High density polyethylene (HDPE)	Usually white colored, high tensile strength. Can stand high temperature and more Durable	Milk and cream bottles, bottles for shampoo and bleach bottles
Low density polyethylene (LDPE)	Soft, flexible plastic	Plastic Bags, Garbage Bags, Meat Plastics, Plastic Cup, Grocery bags, Frozen food bags and computer components
Linear Low density polyethylene (LLDPE)	flexibility and toughness	plastic wrap, stretch wrap, pouches, toys, covers, lids, pipes, buckets, and containers, covering of cables, geomembranes, and mainly flexible tubing.
Expanded polystyrene (EPS)	Foamed, lightweight, energy absorbing, thermal insulation	Hot drink cups, takeout food containers, meat trays, packaging
Polyphenylene sulfide (PPS)	Semi-crystalline thermoplastic, High mechanical strength	Yoghurt containers, plastic cutlery, imitation crystal ‘‘glassware’’

Table 2. Synthesis of related studies

Author	Year	Title	Relevant Findings	Relationship to the Study
G.R. Koerner et. al	2007	The durability of geosynthetics	Polystyrene is an aromatic polymer while styrene is an aromatic monomer.	Both discussed the type of plastic used.
Bjørn Petter Jelle	2016	Nano-based thermal insulation for energy-efficient buildings	Small spheres of polystyrene (from crude oil) with an expansion agent (e.g., pentane C6H12) that expand when heated with water vapor are used to make expanded polystyrene (EPS).	Both used the same type of substance that will be used in the study which is the EPS.
J.M. Khatib et. al	2019	Characteristics of concrete containing EPS	EPS is mostly utilized as a packing or insulating material. It has a low heat conductivity, making it a good insulating and transportable material.	The use of EPS as a packaging material. As well as the discussion about its characteristics.
Geyer et al	2017	Production, use, and fate of all plastics ever made	PE resins account for 34% of the entire plastics market as of 2017, with over 100 million tons manufactured annually	In both studies, the use of polyethylene was discussed.
Fortune Business Insights	2020	Polyethylene Market Size, Share & COVID-19 Impact Analysis, By Type, By End User	The global polyethylene market was valued at $107.43 billion in 2019 and is expected to grow at a CAGR of 3.4 percent to $130.26 billion by 2027.	Both studies discussed the growing percentage of plastics.
T.K. Goswami, S. Mangaraj	2011	Advances in polymeric materials for modified atmosphere packaging (MAP)	HDPE has a higher working temperature than LDPE because of this, but it is also harder, stronger, and slightly heavier (but less ductile).	Both studies discuss the use of High Density Polyethylene.

J. Bayer et. al	2017	Cellulose polymer composites (WPC)	Shampoo bottles, toys, chemical containers, and other common applications include HDPE.	Both studies discuss the applications of HDPE.

Table 3. Synthesis of related studies

Author	Year	Title	Relevant Findings	Relationship to the Study
Shu-Cai Li et.al	2012	Properties of LLDPE Film Modified by LDPE	Linear low density polyethylene (LLDPE) is a copolymer of ethylene and a-olefin with linear sequences and a certain amount of short chain branches responsible for its high crystallinity, endowing LLDPE with superior properties such as tensile strength, tear strength, and puncture resistance, and has thus been widely used in the film industry.	In both studies, the said plastic, Linear low density polyethylene (LLDPE) was used.
H. Li and M. Wang	2021	Electrospinning and nanofibrous structures for biomedical applications	PUs can be either thermoplastics or thermosets. PUs is used to make foams, as well as coatings, adhesives, and varnishes	The discussion of Polyurethane in both studies.
H. Fałtynowicz et. al	2022	Polyurethanes	Because of its biocompatibility, barrier properties, and oxygen permeability, PU has been used in wound dressings.	Both studies show the applications of PU.
S. Park, M. Seo	2011	Element and Processing	PPS does not dissolve in any solvent at temperatures below around 200 °C. PPS is one of the most important high-temperature polymers due to its multiple desired characteristics. These qualities include resistance to heat, acids, and alkalies, as well as mildew, bleaches, aging, sunshine, and abrasion.	Both studies discussed the characteristics as well as the use of PPS.

A.M. Diez-Pascual	2015	Preparation and characterization of polyphenylene sulfide nanocomposites	PPS is a chemically resistant engineering thermoplastic with outstanding mechanical and thermal properties found in the electronics, automotive, aerospace, and chemical industries.	In both studies, the use of PPS (polyphenylene sulfide) was being discussed.

2.2 Organisms

2.2.1 Tenebrio molitor

Mealworms resemble wireworms in appearance and can grow up to 1 ¼ inches in length. They have always been familiar to people because they can be used as food for birds, reptiles, fish, and small mammals. Mealworms are larvae of large black beetles that belong to the genus *Tenebrio,* a latin word meaning "darkness", insects that belong in this genus are nocturnal and are frequently found in dark places [25].

Figure 3. Tenebrio Molitor Larvae

2.2.2 Life Cycle of Mealworms

Females lay their white oval-shaped eggs in small batches. Egg incubation period is greatly influenced by the temperature, the eggs hatch from 4 to 7 days at 26° to 31°C, temperatures lower than 26°C may extend the incubation period up to 19 days.After transforming into the worm or larva, it can be isolated and transform

into pupas.Pupal forms duration range from 6 days at a temperature of 27°C to 18 days at 18°C [26]. Adults first emerge in a whitish color with soft exoskeletons, which in time will gradually harden and darken, soon after mating and oviposition will begin 3 days after emergence [27].

2.2.3 Different Plastic Biodegradation of Mealworms

According to Brandon et al. [9] the survival rate of mealworms fed PE was 98.3% which was not far from the control group (bran fed). The survival rate of mealworms fed PE alone and PE + bran. There was also an increase in the consumption of PS and PE as well as an increase in consumption when co-fed with bran. The addition of small portions of table sucrose (25mg) was able to double the PS consumption of mealworms, however the addition of cinnamon showed no difference in the rate of consumption compared to the no additive control. A plastic only diet did not show any abnormalities during development (larvae to beetle), the frass showed no additional toxic degradation products once filtered, this means that the PS ball fed was safely degraded.

2.2.4 Zophobas morio

Superworms are yellow with dark brown anterior and posterior ends, cylindrical, sclerotized exoskeleton. They can grow up to 55mm in length [7]. They are used as a food source for animals like fish, reptiles, and birds [28]. Superworms belong to the large beetle family of Tenebrionidae, which contains many stored product insect species such as T. molitor and A. diaperinus. Although Zophobas

morio is considered to be among storage insects they have been found in association with only one stored commodity (wheat flour) [7].

Figure 4. Zophobas morio

2.2.5 Life Cycle of Superworms

Zophobas morio eggs are white, oval with round edges that are 1.7mm in length and 0.7mm in width. A female superworm beetle can lay up to 2,200 eggs during its lifespan, with the amount of eggs being positively correlated with adult density. Larvaes hatch after 8 days at 25°. These larvae will fail to pupate under crowded conditions, and larval molting will continue to occur until death. After being isolated for approximately 6 days at 25°C, larvae will become immobilized in a c-shaped posture. The prepupae do not walk, but they can respond by flicking their body, and it takes them 7 more days to become pupae. Pupae have the ability to rotate their abdominal segments in a circular motion as a defense mechanism against predator attacks and larval cannibalistic behaviors. Duration of the pupal stage is 13 to 15 days at 25°C, depending on the pupal weight (adults emerge faster) and temperature (adults emerge faster at 29°C). At this stage individuals can already be sexed by noting two distinct pygopods of the female pupae which are absent

from the male ones. Adults are large in size ranging from 38mm to 57mm with elongated body filiform antennae. Adults can live up to 6 months [7].

Table 4. Life cycle of the superworm

Life Stage	Durations
Eggs	• Will hatch after 8 days at 25°C
Larvae	• 6 months to 1 year life span when kept in groups • 6 days after isolation at 25°C, larvae will start to pupate • It takes 4-5 months for the larvae to become eligible for breeding
Pupa	• Pupal stage is 13-15 days at 25°C (faster adult eclosion at 29°C)
Beetle	• 6 months life span

Table 5. Egg and frass information of the superworms.

Life Stage	Length (mm)	Width (mm)	Size of Mesh Screen Hole (mm)
Egg	1.7	0.7	1.19mm
Frass	<0.595	<0.595	0.595mm

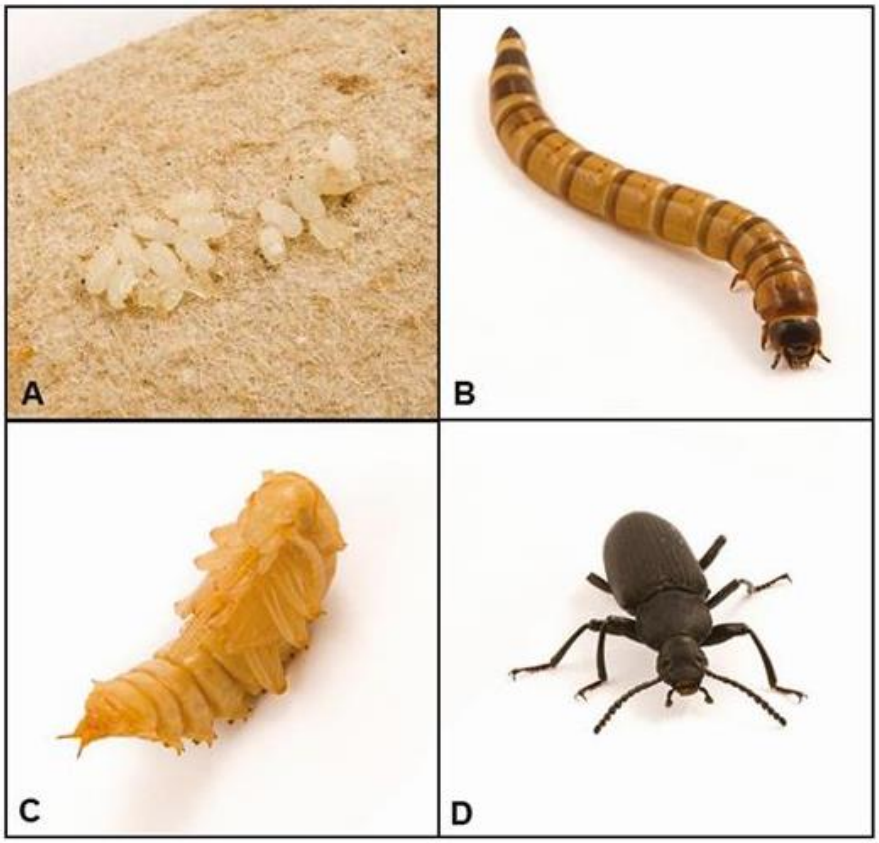


Figure 5. Life stages of superworms

2.2.6 Nutritional value of Superworms

Comparison of four species for commercially available feeder was done by checking the different nutritional content of the following species namely crickets, mealworms, superworm and waxworm. Different results were obtained and it was observed that superworms and waxworms have a high fat and protein content compared to wax worms and crickets. Also it is observed that out of all the species, superworms have vitamin D2 and are a good source of amino acids. Though it is low on calcium content, it can be fed with calcium additives to make the superworms nutritious for intake of different animals [29].

2.2.7 Market Size and Value of Superworms

Marketability of products depend on the supply and demand of it, however regarding with the market size of the superworms, various reports have shown that it has a big market and demand that largely comes from exotic pet owners and even delicacy.In a report featured by the Magazine Agriculture, Peruelo stated that larvae are segregated and sold by weight and size. It spans from the micro worms to jumbo worms, with the smallest that are feeds to small fishes, small worms that are fed to reptiles [30]. From the various exotic pet owners that use superworms as feeds, it can also be a delicacy for some countries and province in the Philippines, a Kuwaiti businessman named Jassem Buabas stated that he can sell the superworms as $3 for 25 pieces and his farm can reproduce 6000 to 9000 superworms per month. This shows that the market of the superworms is highly looked over as it can fetch a high profit depending on the season and location of the seller [31].

2.2.8 Superworm Frass

Superworm frass were used in a study to check its effectiveness as a fertilizer. With the results showing that superworms fed on bran and on plastics had produced similar results and had no changes observed. This showed that superworms can effectively biodegrade plastics as well as help in the plastic economy [32].

2.2.9 Living Conditions

A research done by Kwak et al. [33] argues that the hatchability of Zophobas atratus eggs was significantly influenced by temperature, after achieving

a high 91% hatchability rate at 30°C and a low 6.5% hatchability rate at 25°C. Photoperiod, luminance, and relative humidity had no significant impact on the hatchability of Zophobas atratus eggs. The highest hatchability rate of 93.6% was achieved at 30°C, 12L/12D, 65% RH, and 1800lx. According to Maciel-Vergara et al. [34] the humidity for the larvae rearing should be below 60% as *P. aeruginosa* thrives well in high relative humidity. Cannibalism is a possible entry route for this opportunistic pathogenic bacteria which is detrimental for insect production.

2.2.10 Plastic Biodegradation of Superworms

Peng [35] claims that Zophobas atratus larvae have the capacity of biodegrading both LDPE and EPS, and after performing antibiotic suppression tests, it was found that the biodegradation of LDPE and EPS were gut microbe dependent. Khan, S. [11] summarized the rates of plastic polymer degradation of PE, PS, PVC, and PU from published sources as well as from their own study. From the summary it can be understood that from a single study a Zophobas atratus larvae has reported a 4.3mg/g/d rate for the consumption of PE, and 2.5mg/g/d for the consumption of PS. According to Yang, S. [36] Zophobas atratus larvae are able to chew and penetrate into PP foams as they do with EPS. PP contains only hydrogen and carbon contents, and does not provide necessary nutrition for the superworms, however the addition of wheat bran relieves this constraint, as wheat bran contains all the necessary nutrients for the superworm to complete its life cycle. The Zophobas atratus larvae had a 3.1mg per 100 larvae when fed with PP alone, and a consumption rate of 3.6mg per 100 larvae when fed PP with wheat bran as an additive.

Figure 6. Superworms' degradation of polystyrene

Insects such as superworms prove to be a handful when it comes to degrading plastics, another study showed that *Zophobas atratus* can degrade three types of plastics including polystyrene (PS), polyethylene (PE), and polyurethane (PU) foam. Compared with the control group, PS- or PU-fed larvae had 100 percent survival rates, whereas PE-fed and starving larvae had 81.67 percent and 65 percent survival rates, respectively. Larvae weight was reduced in both the plastics-fed and starving groups. PS, PE, and PU consumption rates were 1.41, 0.30, and 0.74 mg/d/larva, respectively [37].

2.2.10.1 Additives

Over the 35-day period, the PP mass consumed by Z atratus larvae fed with PP alone and PP + wheat bran was 211.1 ± 15.7 mg and 294.9 ± 3.7 mg, implying that feeding co-diet increased PP consumption by 39.7 percent. PP contains hydrogen and carbon elements, but it does not supply enough nutrition (N, P, Na, K, trace elements, amino acids, etc.) for long-term survival and growth. Although digestion of PP provides an energy source for life conditioning and sustains a higher survival rate for larvae fed with PP only than for larvae not fed for a short term (3-5 weeks), the larval weight was reduced due to lack of nitrogen sources and other nutrients. This condition was alleviated with the addition of co-diet WB. The nutrition required by Z. atratus larvae to synthesize enzymes and digestive reagents was obtained from WB. Therefore, the PP consumption activities were improved and resulted in approximately doubled consumption rates compared to PP fed alone [36].

2.2.10.2 Microbiota

Through the analysis of frass using gel permeation chromatography (GPC), solid-state 13C cross-polarization/magic angle spinning nuclear magnetic resonance (CP/MAS NMR) spectroscopy, and thermogravimetric interfaced with Fourier transform infrared (TG-FIR) spectroscopy, the researchers found that the gut microbiota of the superworms is the main reason in degrading plastics, this would be a novel bioresource for pursuit of plastic-degrading enzymes [8]. Some of these plastic degrading gut

bacteria were identified as the Pseudomonas, Rhodococcus and Corynebacterium, that possess genes associated with polystyrene degradation [38]. In an experiment made by Arunrattiyakorn, they isolated the bacterias extracted from the gut of superworms and tested the surface morphology (SEM and WCA analyses) and chemical modification (FTIR analysis) in its PS degradation process [39]. 3 PS-degrading bacteria identified as Pseudomonas sp., Bacillus sp. and Brevibacterium sp. Were used and the Brevibacterium sp. was the most efficient PS-Degrading strain.

Penicillium raperi, Aspergillus flavus, Penicillium glaucoroseum and Pseudomonas sp. were isolated as the most plastic degrading microbes. Among 17 candidates, there was significant growth of 4 strains on plastic films. These strains were identified as Penicillium raperi (derived from AS) grew on PS and PE; Aspergillus flavus (derived from, FS, soil, wax and mealworms' excreta) grew on PE and PS; Penicillium glaucoroseum (derived from AS) grew on PET and PS; and Pseudomonas sp. (derived from AS, FS and soil) grew on PET, PE and PS

Polyethylene (PE), Polyphenylene sulfide (PPS), Polystyrene (PS), Polypropylene (PP) were the plastics that were used in the biodegradation process. P. aeruginosa is the isolated gut bacteria from the superworms, and demonstrated average weight loss of 0.64% per day, 0.53% per day, 0.098% and 0.025% per day to the PE, PPS, PS and PP, respectively [10].

Table 6. Mealworms vs. superworms

Types of Plastics	Yellow Mealworm (Tenebrio molitor)	Superworm (Zophobas morio)
Polystyrene (PS)	1.14 - 1.40mg/g/d [11] 40.89mg/g over 35 days [40] 0.07 mg/g/d [6]	4.49 mg/g/d [11] 60.49mg/g over 35 days [40] 2.78 mg/g/d [6]
Polystyrene + additives (sucrose)	2.54x [41]	1.83x [41]
Polyethylene (PE)	.288mg/g/d [11]	.43mg/g/d [11]
Polyethylene + additives	4.00mg/g/d (bran) [11]	4.43 to 5.25 mg/g/d (cofeed) [11]
Polyurethane (PU)	25.80 mg/g over 35 days [40]	21.06 mg/g over 35 days [40]
Polyphenylene sulfide (PPS)	No data	From 300 μm diameter the weight of PPS decreased by approximately 5.2% within 10 Days. [42]
Polypropylene (PP)	96.1 ± 5.3 mg within 35 days [34]	211.1 ± 15.7 mg within 35 days [34]

Table 7. Synthesis of related studies

Author	Year	Title	Relevant Findings	Relationship to the Study
Cotton, R. T.	1940	Mealworms	Mealworms are larvae of large black beetles that belong to the genus *Tenebrio,* a latin word meaning "darkness", insects that belong in this genus are nocturnal and are frequently found in dark places.	Both studies show the discussion about the mealworm, a larva of black beetles.
Cotton, R. T.	1927	Notes on the Biology of the MealWorms, Tenebrio Molitor Linne and T. Obscurus Fab	After becoming a full grown larva, it may transform to the pupal form.Pupal forms duration range from 6 days at a temperature of 27°C to 18 days at 18°C	Both Studies discuss the life cycle of the mealworm
Ribeiro, N., Abelho, M., & Costa, R.	2018	A Review of the Scientific Literature for Optimal Conditions for Mass Rearing Tenebrio molitor (Coleoptera: Tenebrionidae)	Adults first emerge in a whitish color with soft exoskeletons, which in time will gradually harden and darken, soon after mating and oviposition will begin 3 days after emergence	In both studies, the evolution of mealworm into beetle was discussed.
Brandon et al	2018	Biodegradation of Polyethylene and Plastic Mixtures in Mealworms (Larvae of Tenebrio molitor) and Effects on the Gut Microbiome	The survival rate of mealworms fed PE was 98.3% which was not far from the control group (bran fed).	The studies show how the organisms can effectively biodegrade Polyethylene.
Rumbos, C. I., & Athanassiou, C. G.	2021	The Superworm, Zophobas morio (Coleoptera:Tenebrionidae): A 'Sleeping Giant' in Nutrient Sources	Superworms are yellow with dark brown anterior and posterior ends, cylindrical, sclerotized exoskeleton. They can grow up to 55mm in length	Both studies provide discussion about the Superworms.
Kwak et al.	2021	Optimal hatching conditions of Zophobas atratus (Coleoptera: Tenebrionidae) eggs under various culture conditions	The hatchability of Zophobas atratus eggs was significantly influenced by temperature, after achieving a high 91% hatchability rate at 30°C and a low 6.5% hatchability rate at 25°C	The life cycle of Zophobas atratus was discussed in both studies.

Table 8. Synthesis of related studies

Author	Year	Title	Relevant Findings	Relationship to the Study
Maciel-Vergara et al	2018	Cannibalism as a Possible Entry Route for Opportunistic Pathogenic Bacteria to Insect Hosts, a Pathogen of the Giant Mealworm Zophobas morio	The humidity for the larvae rearing should be below 60% as *P. aeruginosa* thrives well in high relative humidity	Both studies discuss how cannibalism occurs in the environment of the superworms.
Peng et al.	2020	Biodegradation of low-density polyethylene and polystyrene in superworms, larvae of Zophobas atratus (Coleoptera: Tenebrionidae): Broad and limited extent depolymerization	Zophobas atratus larvae have the capacity of biodegrading both LDPE and EPS, and after performing antibiotic suppression tests, it was found that the biodegradation of LDPE and EPS were gut microbe dependent.	Both studies show the ability of the Zophobas atratus to biodegrade plastics such as LDPE and EPS.
Khan, S. et a.l	2021	Valorizing plastic waste by insect consumption	Summarized the rates of plastic polymer degradation of PE, PS, PVC, and PU from published sources as well as from their own study.	In both studies, the consumption rate of different plastics by Superworms were discussed.
Yang, S. et al.	2021	Biodegradation of polypropylene by yellow mealworms (Tenebrio molitor) and superworms (Zophobas atratus) via gut-microbe-dependent depolymerization	Zophobas atratus larvae are able to chew and penetrate into PP foams as they do with EPS.	The biodegradation of polypropylene was shown in both studies.
Yang, Y. et al.	2020	Biodegradation and mineralization of polystyrene by plastic-eating superworms Zophobas atratus	Researchers found that the gut microbiota of the superworms is the main reason in degrading plastics, this would be a novel bioresource for pursuit of plastic-degrading enzymes.	Both studies discuss that the gut microbiota of the superworms is the main reason for degrading plastics.
Sun, J. et al.	2022	Insights into plastic biodegradation: community composition and functional capabilities of the superworm microbiome	Some of these plastic degrading gut bacteria were identified as the Pseudomonas, Rhodococcus and Corynebacterium, that possess genes associated	Pseudomonas sp. was discussed in both studies, a bacteria that can degrade polystyrene.

		in styrofoam feeding trials	with polystyrene degradation	

Table 9. Synthesis of related studies

Author	Year	Title	Relevant Findings	Relationship to the Study
Arunrattiyakorn, P. et al.	2022	Biodegradation of polystyrene by three bacterial strains isolated from the gut of Superworms (Zophobas atratus larvae)	They isolated the bacterias extracted from the gut of superworms and tested the surface morphology (SEM and WCA analyses) and chemical modification (FTIR analysis) in its PS degradation process	Both studies showed that the bacteria from the gut of superworms can help in the biodegradation process of polystyrene.
Lee, H. M. et al.	2020	Evaluation of the Biodegradation Efficiency of Four Various Types of Plastics by Pseudomonas aeruginosa Isolated from the Gut Extract of Superworms	P. aeruginosa is the isolated gut bacteria from the superworms, and demonstrated average weight loss of 0.64% per day, 0.53% per day, 0.098% and 0.025% per day to the PE, PPS, PS and PP, respectively	Both studies show that Pseudomonas sp. Bacteria from the gut of superworms can degrade different kinds of plastics such as PE, PS, and PP.
Kuan, Z.-J. et al.	2022	Worming the Circular Economy for Biowaste and Plastics: Hermetia illucens, Tenebrio molitor, and Zophobas morio	The GC-MS library search result did not find major toxic degradation products in the filtered worms frass	Both studies show that having plastics as a sole diet of superworms has no effect in their biowaste.
Tan, Y.	2021	Former BPO employee finds success in farming insects for pet food.	The market size and value of the Superworms	Both studies shows that Superworms is a good business.
Lebel, S.	2021	Kuwaiti breeder eyeing superworms as new superfood.	The production, market size, and value of the Superworms.	Both studies shows how high in demand superworms are in owners of birds, fish, amphibians, and reptiles.

2.3 Monitoring system

2.3.1 raspberry pi camera

The credit card size Raspberry Pi (RPI) with Open Source Computer Vision (OpenCV) software handles the image processing, control algorithms for the functions and sends captured pictures to the user via Wi-Fi. [43].

2.3.2 YOLOv4 Machine Learning Algorithm

The you only look once version 4 (YOLO v4) object detection network is a one-stage object detection network and is composed of three parts: backbone, neck, and head. The backbone is a pretrained CNN such as VGG16 trained on COCO data sets. The neck connects the backbone to the head and it is composed of a spatial pyramid pooling module and a path aggregation network. These features help in mapping the different layers of the backbone network and send them as inputs to the head. The head processes every input and provides predictions regarding the object detection and tracking [44].

Table 10. Synthesis of related studies

Author	Year	Title	Relevant Findings	Relationship to the Study
Abaya, W. F., Basa, J., Sy, M., Abad, A. C., & Dadios, E. P.	2014	Low cost smart security camera with night vision capability using Raspberry Pi and OpenCV	The credit card size Raspberry Pi (RPI) with Open Source Computer Vision (OpenCV) software handles the image processing, control algorithms for the functions and sends captured pictures to the user via Wi-Fi.	Both studies used camera as it has the ability to process images in lowlight.
Bochkovskiy, Alexey, Chien-Yao Wang, & Hong-Yuan Mark Liao	2020	YOLOv4: Optimal Speed and Accuracy of Object Detection	The you only look once version 4 (YOLO v4) object detection network is a one-stage object detection network and is composed of three parts: backbone, neck, and head.	Both studies used YOLO as its machine learning.

2.4 Related Studies

2.4.1 PolyWorm

For the Version 1, entitled “Semi-Automated Biodegradation System for Polystyrene-Eating Mealworms with ESP8266 for Wireless Notification System”, it was designed with a Semi-automatic mealworm (T. molitor) life stages segregation using Mesh Screen and vibration motors [45]. This was accompanied by a Notification system using Blynk software for the notifications of the weight of the gain product (frass), manual segregation and collection of the mealworms and its life stages. As well as an automatic ventilation system to maintain the ideal living

conditions of the organism. This project aims to biodegrade Expanded Polystyrene (EPS) and Extruded Polystyrene (XPS).

2.4.2 PolySeg

Version 2 of the "Semi-Automated Biodegradation System for Polystyrene-Eating Mealworms with ESP8266 for Wireless Notification System", is now improved with the projected entitled "Fully Automated Polystyrene Biodegrading System for Mealworm Segregation using Arduino Mega 2560 with ESP32 for Wireless Notification". It has the same components and processes as the Version 1, the main difference is that the mealworm segregation process is fully automated compared to the previous manual collection and segregation of the organisms. It also uses a website rather than an application and the plastic to be biodegraded is EPS only [46].

2.4.3 SuperB

Version 3 took a whole new approach in the biodegradation of plastics. This system uses Superworms (Zophobas morio) as the organisms that biodegraded plastics. It has the same hardware components as its predecessors, the main differences are the utilization of a web camera for the remote monitoring system. The use of YOLOv4, a computer vision that counted the organisms to ensure that the system helps in giving an ideal living environment for the superworms, and an automated hatching and breeding system for Superworms. The plastics to be biodegraded are Polystyrene(PS), Expanded Polystyrene (EPS), Polyethylene (PE), and Polyurethane (PU).

Table 11. Comparison of version 1, 2, and 3 researches

	Version 1	Version 2	Version 3
Title	Semi-Automated Biodegradation System for Polystyrene-Eating Mealworms with ESP8266 for Wireless Notification System	Fully Automated Polystyrene Biodegrading System for Mealworm Segregation using Arduino Mega 2560 with ESP 32 for Wireless Notification	Automated Superworm Segregation System for Multi-Plastic Biodegradation using ESP32 with Computer Vision for Real-Time monitoring via Mobile Application
Problem	Biodegradation of Polystyrene	Biodegradation of Polystyrene	Biodegradation of Different plastics
Solution	Semi-automated biodegradation system for mealworm with alert system	Fully-automated biodegradation system for mealworm with alert system	Fully-automated multi-plastic biodegradation system and ecosystem for Superworm with real-time monitoring using Computer vision camera and IoT with alert system
Technologies Involved	● Semi Automatic mealworm life stages segregation ● Uses Mesh screen and vibration system ● Notification system via Blynk ● Sequence and Pre Scheduled setup based operation ● Automatic Ventilation system ● Measures the weight of the gain product (frass) and notifies the user ● Time based food refill ● Internet of Things based ● Mealworm	● Automatic Mealworm segregation system ● Uses Mesh screen and vibration system ● Full control and manual override using website via esp 32 ● Automatic Ventilation system ● Weight based notifications for food refill or overpopulation ● IOT based ● Mealworm (*Tenebrio molitor*)	● Automatic Superworm segregation system ● Automated Hatching and breeding system ● Uses mesh screens, vibration motors and opening mechanisms for segregation ● Automatic Ventilation system ● Full control and manual override using application via Thunkable ● Full control and manual override using physical buttons ● Computer Vision YOLOv4 ● Automatic feeding system ● Iot Based ● Superworm (*Zophobas morio*)

	(*Tenebrio molitor*)		
Plastics for biodegradation	Expanded Polystyrene (EPS) Extruded Polystyrene (XPS)	Expanded Polystyrene (EPS)	Polystyrene(PS) Expanded Polystyrene (EPS) Polyethylene (PE) Polyurethane (PU)

Table 12. Synthesis of related studies

Author	Year	Title	Relevant Findings	Relationship to the Study
Alonzo, R. et al.	2020	Semi-Automated Biodegradation System for Polystyrene-Eating Mealworms with ESP8266 for Wireless Notification System	Pursues to biodegrade polystyrene with the use of mealworms by providing a system that continually segregates and provides artificial habitat for them to develop from one stage to another and then to reproduce.	Both studies created a system that will help to biodegrade polystyrene.
Migullas, R. et al.	2020	Fully Automated Polystyrene Biodegrading System for Mealworm Segregation using Arduino Mega 2560 with ESP 32 for Wireless Notification	Aims to develop a fully automated device that will biodegrade polystyrene through the utilization of yellow mealworms. The control system of the device will be based on the integration of Arduino Mega 2560 for the automation of segregation module, collection module and food dispensing	Both studies involve the creation of a fully automated device that will biodegrade polystyrene. As well as to utilize the use of the organism.

2.5 System Design

2.5.1 Hatching and Breeding container

In the breeding process, it is necessary for the device to have an individual compartments to meet the conditions for the larva to pupa stage. Experiments show that 4-6 days is the period needed for the metamorphosis to come through the modulation of juvenile hormone titres [47].

In superworms, the crowding of superworms inhibits pupation, whereas the isolation of a single superworm allows the pupation stage. Allowing the life cycle of the organism to take place [48].

2.5.2 Linear Actuator

The automation of the mechanisms of the device will be done with the use of linear actuators as it will be in control of the moving parts of the system, namely the containers and the different connectors in between containers.

The main motivation for determining efficiency is that the linear actuator contains many small moving parts that are interconnected with one another. The tests went above and beyond, with actuators tested to their maximum rated load limit. The actuators were put through their paces with a variety of loads and input voltages. The results of the tests conducted showed that the actuators were pushed to their limits while still ensuring that they’ve performed their duty reliably. [49].

2.5.3 Segregation of Superworms using Mesh Screen and Vibrating Motor

Different processes and techniques are used in the mining industry to extract minerals from the ground. Shovels, pickaxes, chisels, hammers, and other tools are used in mining. The mining industry's machinery is still evolving today. With the technological advancements and innovations that are being applied in this industry, the extraction of minerals became more developed by using vibrating screen machines. Because of this screen machinery, it provided a way in the segregation of the organisms [50].

The mesh used in the different containers will be the filter to separate the organisms and its frass. The other mesh screens will be used as nets in the segregation of the life stages of the worms, different gauge sizes will be used depending on the life stages [50].

2.5.4 Pulley System using DC Motor

Worm gearboxes are commonly used when compactness and orthogonal redirection are important considerations. One key advantage of these gears is their inherent self-locking capability, meaning that rotation can only occur through the input side avoiding the side with the load [51].

2.5.5 Temperature-Based Automatic Fan Speed Controller

Temperature is one of the most important living conditions of this ecosystem. This will be controlled by a temperature-based Automatic Fan Speed Controller. A DHT11 Temperature sensor that can sense both temperature and

humidity was used. It was programmed using Arduino UNO and the fan speed was controlled using the technique of Pulse Width Modulation (PWM) based on the temperature sensed by the temperature sensor [52]. Controlling and maintaining the temperature will be necessary to achieve an efficient biodegradation system [53].

2.5.6 Automatic Pet Feeder

The unsupervised feeding of pets is a necessary innovation that can be applied to this study. A fully automated device needs automation in food distribution to have an efficient biodegradation system.

For this reason, an automatic pet feeding system was developed to ensure that pets can get food at a specific interval of time. The machine is connected to a real-time clock for real-time monitoring and is integrated with a servo motor for the delivery of food from the storage to the feeding bowl. For the automatic controlling of the machine, an Arduino UNO was used [54].

2.5.7 Notification System (THUNKABLE)

The system will use thunkable, a platform used to create mobile applications. This platform will be used for the notification system. A study entitled 'Monitoring System for Broiler Chicken Farms Based on Internet of Things' conducted in an Indonesian Computer University shows that thunkable is able to create a notification system. Thus the data that the researchers gathered can be accessed in real-time using a smart phone application [55]. Furthermore, a research

from the journal: Analytical Methods, used push notification to assist and evaluate students, thunkable was also used in making this study [56].

2.5.8 Google firebase

Google Firebase is a real-time database that is used for automation and IoT projects. This new technology is free and is designed to be as a general database with little to no expertise needed [57].

2.5.9 Thunkable

In developing the application, Thunkable will be utilized to perform the back-end and front-end functionalities of the software. Thunkable's internet platform is a simple tool for developing mobile apps. Computational thinking is at the foundation of problem solving in computer science, and research reveals that when students utilize simple block coding systems similar to the style used for Thunkable, their computational thinking improves [58]. One significant benefit of this is that individuals may look at other people's work, examine how they ordered their components, and adapt other people's code to match their own needs.

2.5.10 Arduino IDE

Arduino IDE is the open-source software that makes it easy in writing codes and uploading it to the arduino boards of a user. This software can be used with any Arduino board. The term "IDE" refers to an official program introduced by Arduino.cc that is mostly used for editing, compiling, and uploading code to the Arduino Device [59].

Table 13. Synthesis of related studies

Author	Year	Title	Relevant Findings	Relationship to the Study
Qeunnedey, A., Aribi, N., Everaerts, C., Delbecque, J.P.	1995	The durability of geosynthetics	This study determined the length of the breeding process and hatching process of the superworms.	This affects in designing the prototype for the habitat of the organisms.
Pinto, V. H., Gonçalves, J., & Costa	2021	Model of a DC Motor with Worm Gearbox	By using a DC motor in conjunction with a worm gearbox, it is possible to ensure that, for example in a robotic manipulator, when a joint in the arm reaches a desired angle, it remains in place until the next required setpoint is reached.	Both studies used DC motor for one of the prototype's mechanism
Makinde, O.	2014	Functionality Assessment of a Reconfigurable Vibrating Screen	This study explains the functionality of using a vibrating screen in segregating objects with different sizes	This will be the mechanism in segregating the organisms in the ecosystem.
Nigade, A.S., et al.	2016	Temperature Based Automatic Fan Speed Controller	This study showed how to make an automatic fan speed controller that adjusts its speed according to the temperature of its environment.	Basis for automating the control of temperature to ensure ideal living conditions
Tiwari, M., et al.	2018	Automatic Pet Feeder Using Arduino	This study showed a technique in automating the pet feeder using arduino and servo motors.	This will be the inspiration for automating the distribution of food to the organisms.

Mamta, Paul, A., Tiwari, R.	2022	Smart Home Automation System Based on IoT using Chip Microcontroller	This study compared the different available IoT microcontrollers and showed that ESP32 is the best IoT microcontroller to be used in an automation system.	This will be the proof and inspiration in making the device into an IoT based ecosystem.
Khawas, C., & Shah, P.	2018	Application of Firebase in Android App Development-A Study	Firebase is a relatively new technology for handling large amounts of unstructured data.	Both studies used Firebase as a database since it can handle large amounts of data.
Siegel, D..	2020	There's an app for that, and I made it	Thunkable's internet platform is a simple tool for developing mobile apps	This will be used for android development.
Fezari, M., & Dahoud, A. Al.	2018	Integrated Development Environment "IDE" For Arduino	mostly used for editing, compiling, and uploading code to the Arduino Device.	It shows how Arduino IDE was utilized in both studies.

Chapter 3

Methodology

This chapter presented the methodology of the study. The proposed methodology was described in the research design, followed by the process flow, and steps in the hardware and software development, establishment of the business model, comparative analysis between the previous versions, and provided the user acceptance rating for the prototype. The Project work plan was also presented to show the goals that were to be achieved per given time.

3.1 Research design

This research used two research designs namely, descriptive and developmental research design. The product of this research was a prototype concerning the automation of the breeding and segregation of the Superworms, as well as the biodegradation of the

different plastics. Thus, making it a developmental research. The prototype automated these processes which required a User Acceptance Test to ensure the quality and efficiency of the product, making the research a descriptive research as well.

Figure 7 shows the Input-Process-Output of the research:

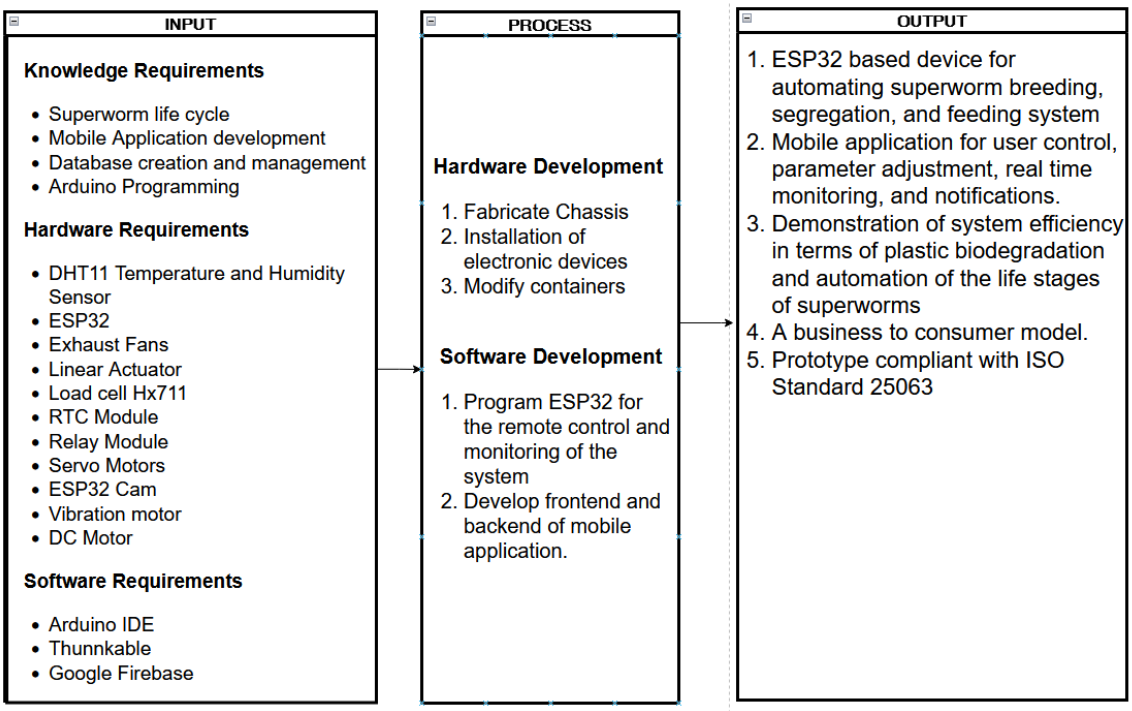


Figure 7. Input process output diagram of the research study

The IPO depicted that the researchers gathered the required background knowledge for the handling of the organisms, programming languages to be used, and application and construction of the hardware and software in the system.

During the development, calibration of sensors, and construction of hardware and software were done. This was carefully monitored throughout the project development to ensure the expected outcome.

The expected output had to be aligned with the objectives stated in the paper, and this was examined and checked after the development period.

3.2 Research Process Flow

The process flow of the research will be illustrated in figure 8 below:

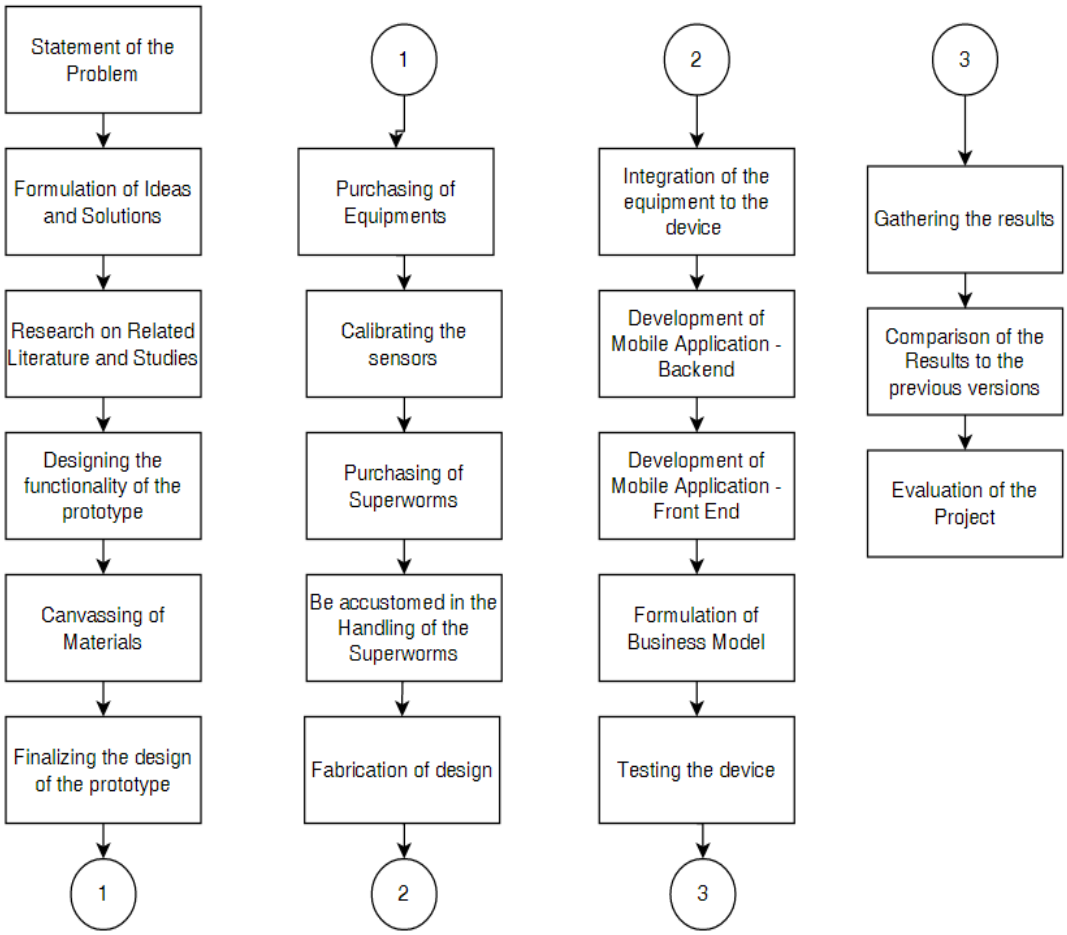


Figure 8. Research Process Flow

3.3 Development of an Automated Superworm Breeding and Segregation device through Integrated Sensors and Equipment

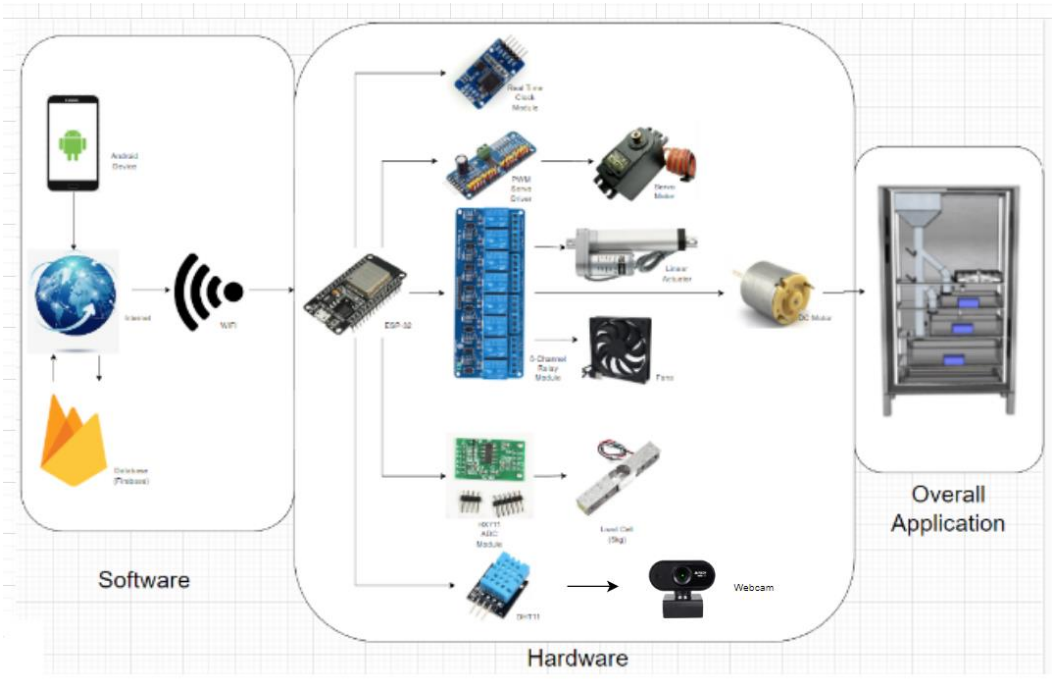


Figure 9. One line diagram of the automated plastic biodegradation for superworm segregation system

3.3.1 Design and Construction of the Food Processing System

For the food processing system, different mechanisms were used to provide food for the Superworms. Different technologies were used to create a functioning food processing system.

The food processing system consisted of 4 different levels, each with its own function. The creation of the mechanisms namely the Tube Switch, Stopper, and the Weight switch was discussed in this part. With this mechanism, the

researchers were able to distribute a specific amount of plastic for a desired compartment.

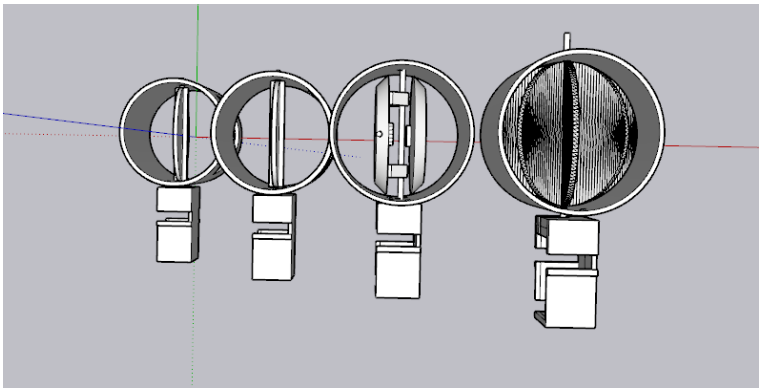


Figure 10. Food processing system flow

3.3.1.1 Design and Construction of the Tubes

For the construction of the food chamber, first the dimensions of the pipes and the fructum is needed. It is also necessary to take into consideration the placement of the joints in each of the compartments where food is needed. For the length and dimensions of the pipes, refer to Table 14.

Table 14. Dimension and measurements of the Food Processing System

Part	Length (in)	Width (in)	Height (in)	Diameter/Diameter Hole (in)
Frustum	11.8	8.3	3..9	none
Extension of Frustum	11.8	8.3	4.5	-
Main Pipe	-	-	30.8	2.6
Beetle Tube (diagonal)	6	-	7.6	2.6
Beetle tube (lid)	6	-	4.9	2.6
Biodegradation Tube (diagonal)	-	-	3	2.6
Biodegradation tube (lid)	-	-	4.2	2.6

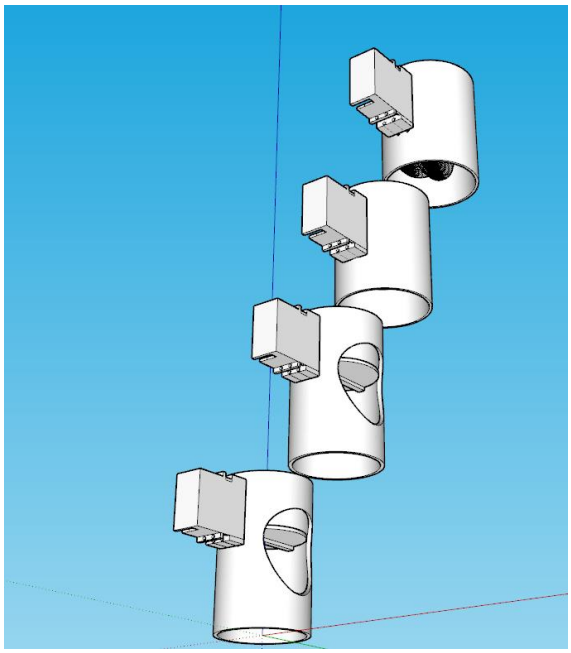


Figure 11. Tubes of the system's food processing

3.3.1.2 Design and Construction of the Tube Switch Mechanism

The Tube Switch mechanism has 5 parts, a 3D modeled tube, a 180 degrees servo motor, a 3D modeled stopper, a M3 sized long screw and a 3D modeled servo motor holder.

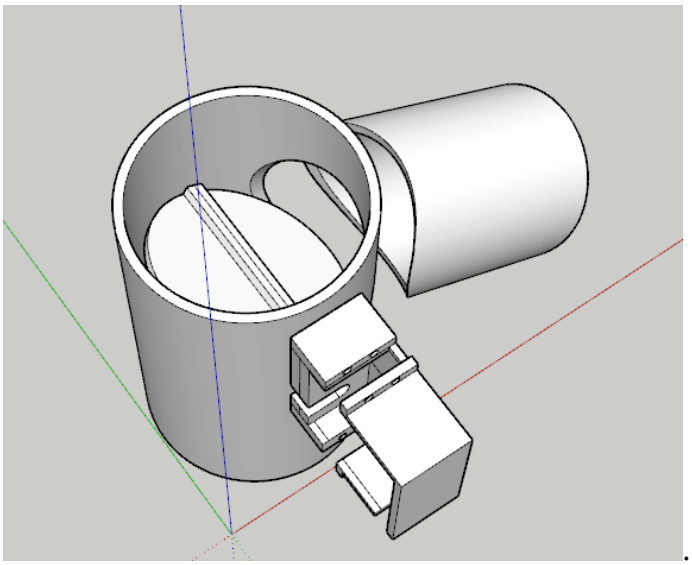


Figure 12. Tube Switch Mechanism

By utilizing the rotating motion of the servo motor, a directional-based switch could be created by allowing the plastics to free fall and slide onto the tube which acted as the switching mechanism of the Food Mechanism. The code of the servo motor was set to activate to 90 degrees if a certain parameter was achieved and would return to its former state if not.

3.3.1.3 Design and Construction of the Stopper Mechanism

The Stopper mechanism has 4 parts, a 3D modeled spherical stopper, a 360 degrees servo motor, a M3 sized long screw and a 3D modeled servo motor holder.

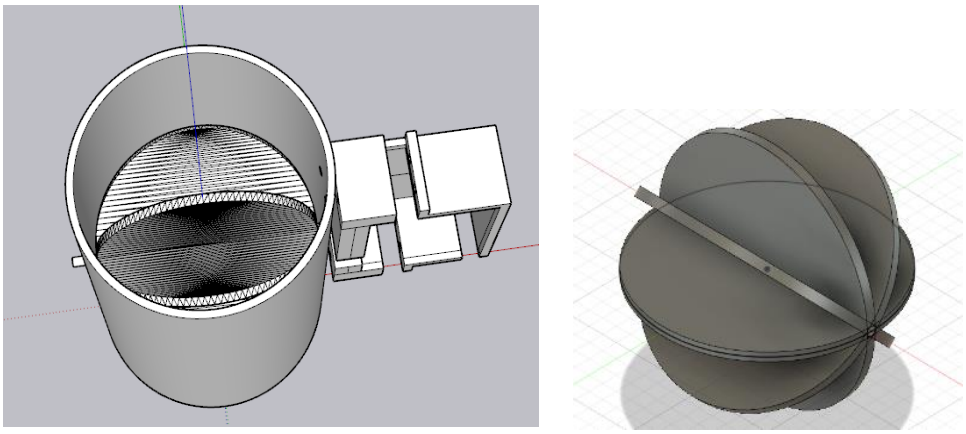


Figure 13. Stopper Mechanism

This part of the system acted as the stopper and the main distributor of food in the food processing system. The servo motor was programmed such that it would activate if the superworms needed food.

3.3.1.4 Design and Construction of Weight Switch Mechanism

The weight switch mechanism has 5 parts, a 3D modeled weight attachment for load cell, a 180 degrees servo motor, a load cell, a M3 sized long screw and a 3D modeled servo motor holder.

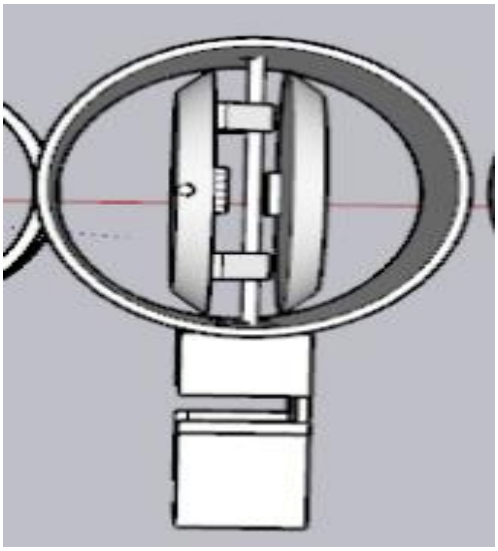


Figure 14. Weight Switch Mechanism

The weight switch mechanism used a load cell and a servo motor as its key functionality. When a certain amount of weight was measured by the load cell, the servo motor would rotate 90 degrees, and after some time, the servo motor would go back to the original state. An interrupt was used in this mechanism so that when the servo motor moved 90 degrees, the measurement of the load cell would stop and resume only when the servo motor went back to its normal state.

The circuit of the load cell was constructed using a HX711 adc module and a 5kg load cell.

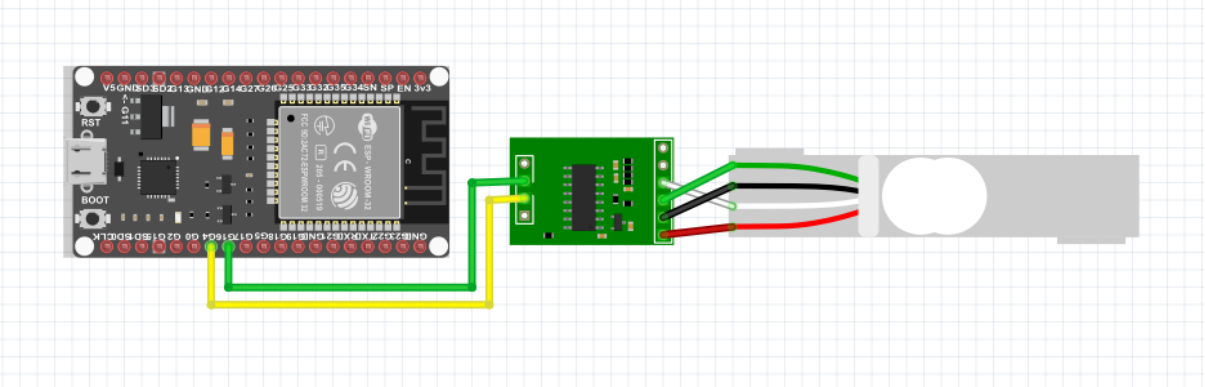


Figure 15. Circuit connection of HX711 and a 5kg load cell.

Load Cell HX711 functioned as the main sensor for the measurement of the weight of the different parts of the system. The following table below represented the number of plastics that were fed to the superworms. Each of the weights was set as the trigger for the weight switch mechanism to work.

Table 15. Sample data comparison between the weighing scale and load sensor.

	WEIGHING SCALE	LOAD SENSOR
TEST NO. 1		
TEST NO. 2		
TEST NO. 3		

Table 16. Weight per container of plastics to be used

Week	Plastics	Weight per container (weekly supply)
1-2	Polystyrene	75 g - Biodegradation container 10 g - breeding container 30 g - egg-small larva container
3-4	Expanded Polystyrene	75 g - Biodegradation container 10 g - breeding container 30 g - egg-small larva container
5-6	Polyethylene	7.5 g - Biodegradation container 3 g - breeding container 5 g - egg-small larva container
7-8	Polyurethane	50 g - Biodegradation container 7.5 g - breeding container 20 g - egg-small larva container

For the functionality of the load sensors to work, all were connected to the ESP32 board and were programmed using the Arduino IDE. In programming the load sensor mechanisms, the HX711 library was used to set the thresholds and mechanisms of the load sensor circuit.

The circuit of the servo motor of each mechanism was connected to a PWM servo motor driver controller which was controlled by the ESP32. The diagram below showed the connection.

The servo motor was used as the moving tool to open each chamber to change the direction of the falling food. By utilizing the rotational mechanism of a servo motor, pulling and pushing could be automated. The researchers made a small cut in the pipe of the food system and installed the servo motor with a sliding mechanism to either block or allow the food to go through. The trigger for the system to allow the food was based on the amount of food per container. If the food was less than 50% of its amount,

it automatically distributed the food using the openings of the Food chamber.

For the programming of the servo motor, the Servo arduino library was required and to test the functionality, Figure 16 was used.

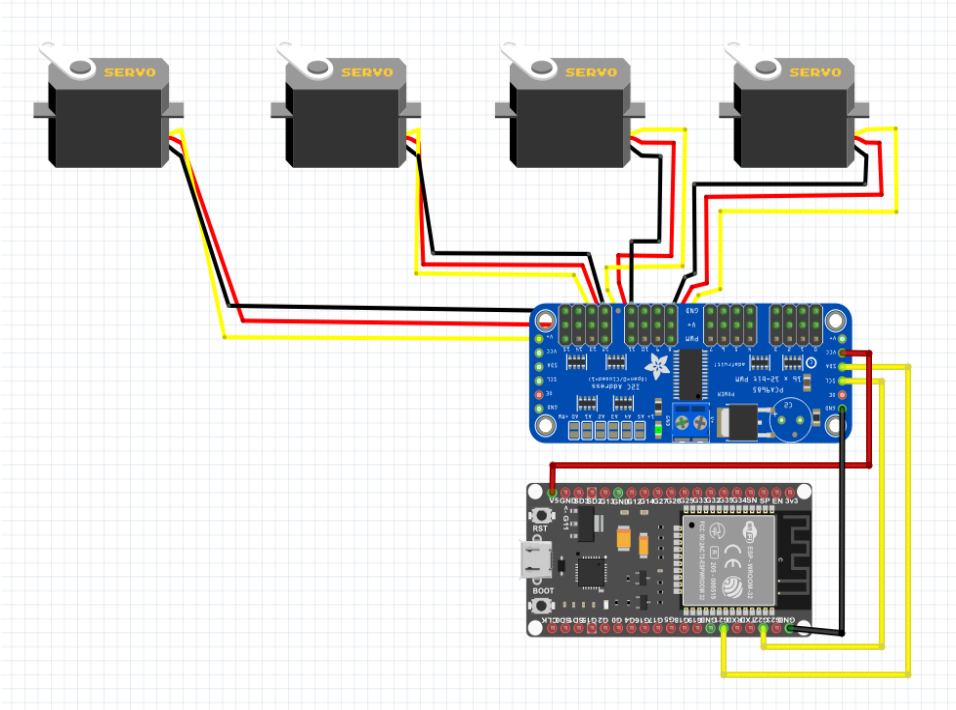


Figure 16. Circuit diagram of the servo motor

After the configuration of the code, the installation of the servo motor was placed in the joints of the pipes of the food chambers to manage the switching of the entrance. There was a spherical plate in between the sections of the pipe that would open or close depending on the weight of the containers. The plate would rotate in a 180-degree manner to ensure that the plastics were redirected to the specific containers.

3.3.2 Design and Construction of the Breeding and Segregation System

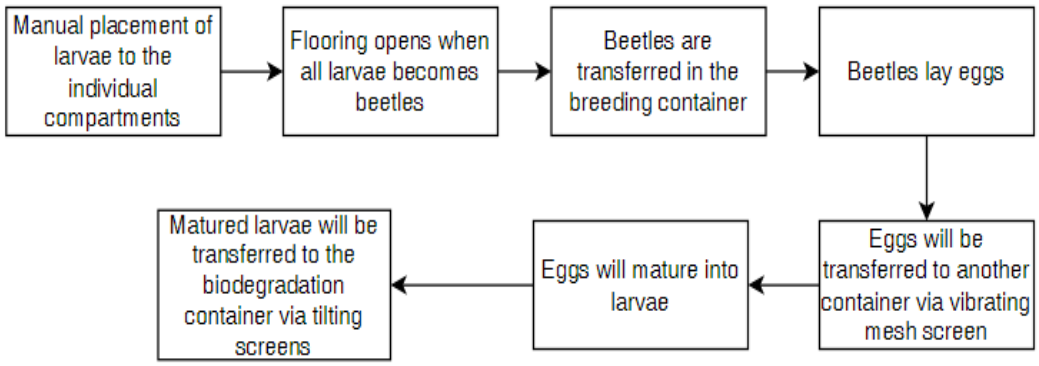


Figure 17. Flow of the breeding and segregation system

For the breeding and segregation system, the design used individual compartments for the larva-pupa-beetle process or the metamorphosis process.

Isolation was needed for the Superworms for them to become beetles, which was necessary in the breeding process. After waiting for 2-4 weeks for the larvae to become beetles, the flooring of the individual compartments opened and transferred the beetles to the breeding container.

In the breeding container, it had a 1/16 in. mesh screen as a flooring with 1.19 mm as the size of its holes. This mesh screen was used for the sifting process for the eggs to be transferred to the container below. After segregating the eggs from the beetles, the system segregated these organisms every 7 days. In transferring the matured larvae into the biodegradation container, another mesh screen acted as the flooring and was opened to slide the larvae into the biodegradation container. The old batch of Superworms was then harvested, depending on the number of new Superworms to be added or by the necessary harvest.

For segregating the frass, another 1/30 in. mesh screen with 0.595 mm as the size of its holes was used. It was used for sifting the frass to be transferred to the frass bin, for the user to collect and sell or be used as a fertilizer. This was done twice a month.

3.3.2.1 Pupation chamber

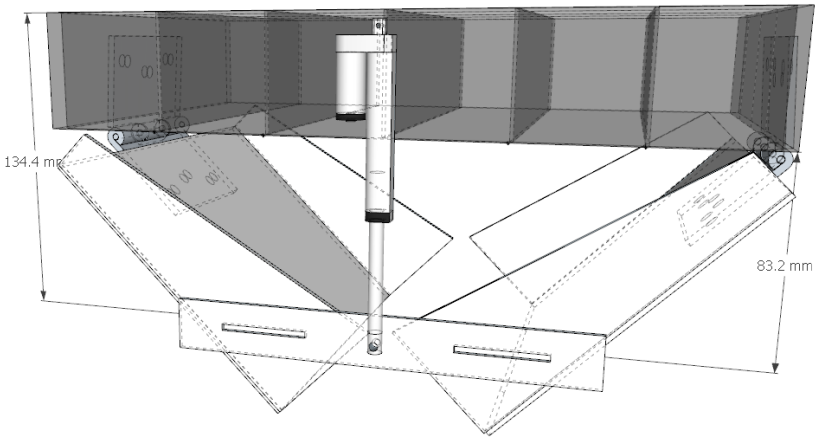


Figure 18. Open view of the pupation chamber

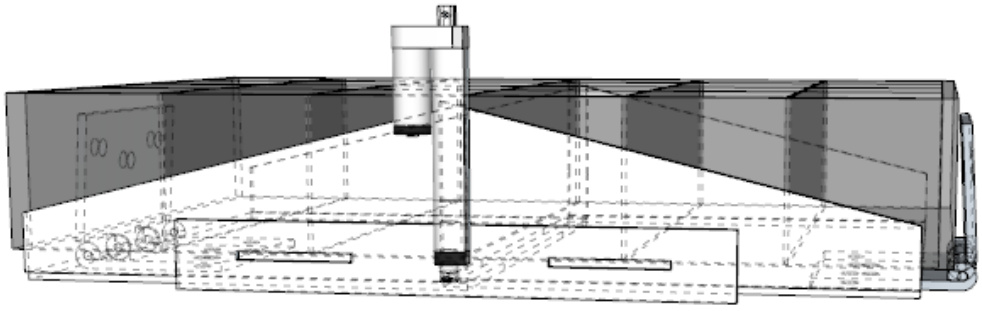


Figure 19. Closed view of the pupation chamber

The pupation chamber was used for the first level of the breeding process, which was necessary for the larvae to pupate and become beetles since superworms needed to be isolated to pupate. The user placed worms

in the different compartments, and the releasing of the beetles was time-based with the use of an RTC module. After 14 days, the flooring of the compartments was opened. The opening mechanism utilized linear actuators and door hinges to achieve a push-and-pull floor opening mechanism. For the selection of a linear actuator, the length of the stroke was dependent on the depth of the opening of the flooring. Since the depth of the flooring was about 83.2 mm, a 12V DC linear actuator with a stroke length of 100 mm was used as the mechanism for opening the flooring.

The circuit of the linear actuator used a 2 relay module which was programmed to work in a pull and push motion. The figure below shows the circuit diagram of the linear actuator connected to the esp32 board.

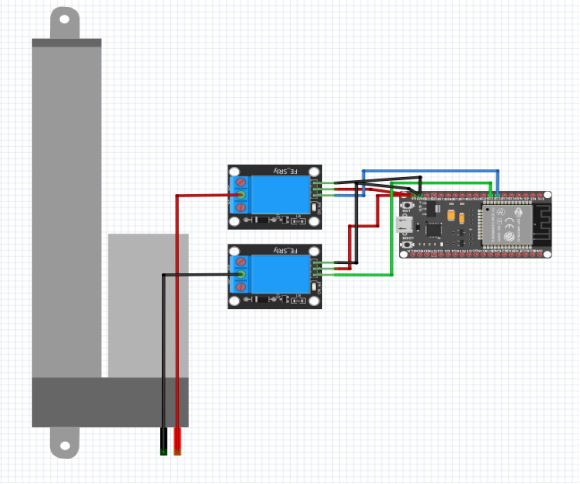


Figure 20. Circuit connection of the Linear Actuator.

The measurements and dimensions of the isolation compartment were tabulated in Table 17.

Table 17. Measurement and dimensions of the isolation compartment

Part	Length (in)	Width (in)	Height (in)
Main compartment	12	8	2
Opening floor	5.85	8.16	-

3.3.2.2 Breeding Container

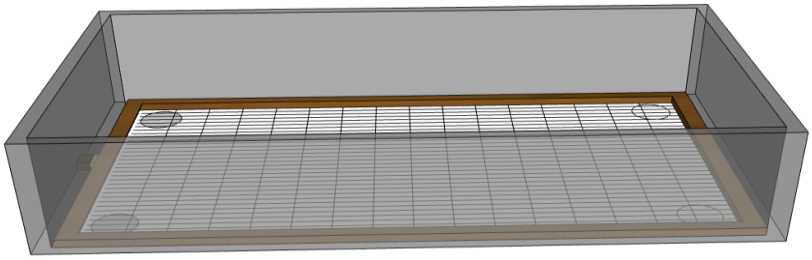


Figure 21. Breeding container

The breeding container was the second level of the breeding system. It housed the beetles that came from the isolation compartment and facilitated the breeding process of the organisms. The trigger for the system to know when the beetles laid eggs was time-based, with the vibration motor functioning for 7 days. Since it took 8 days for the eggs to hatch [7], a 1/16 in. mesh screen was used as the flooring. The measurements and dimensions of the breeding container were presented in Table 18.

Table 18. Measurement and dimensions of the breeding container

Part	Length (in)	Width (in)	Height (in)
Main compartment	15.3	7.7	3.1
Opening floor	16	8.3	3.1

The breeding container had several vibration motors attached to it. The vibration motors acted as the sifting mechanism for the frass produced by each superworm. 4 vibration motors were attached to each corner of the mesh screen to provide equal vibration to the whole mesh screen. The circuit for the mesh screen was provided in Figure 22.

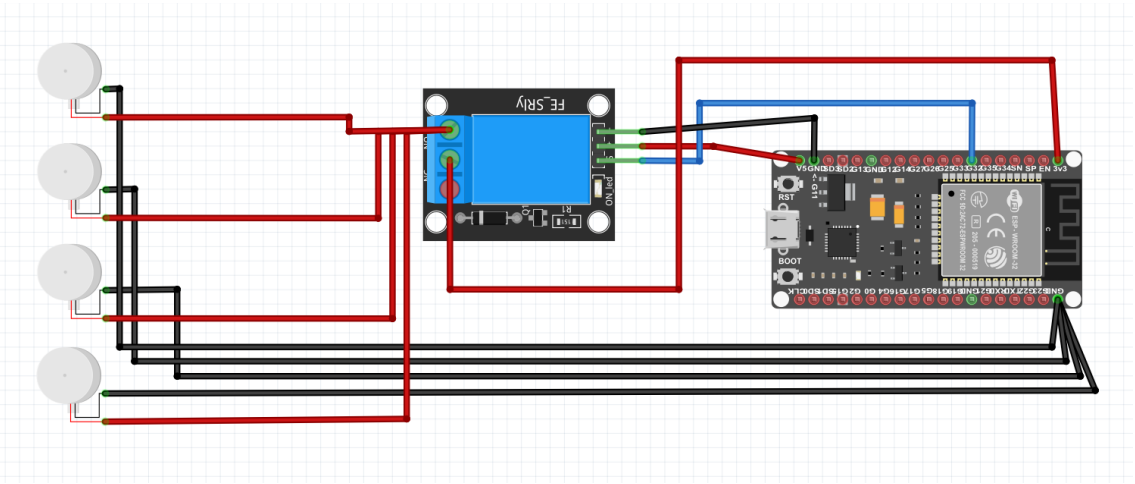


Figure 22. Circuit diagram for the Mesh Screen**.**

3.3.2.3 Hatching Container

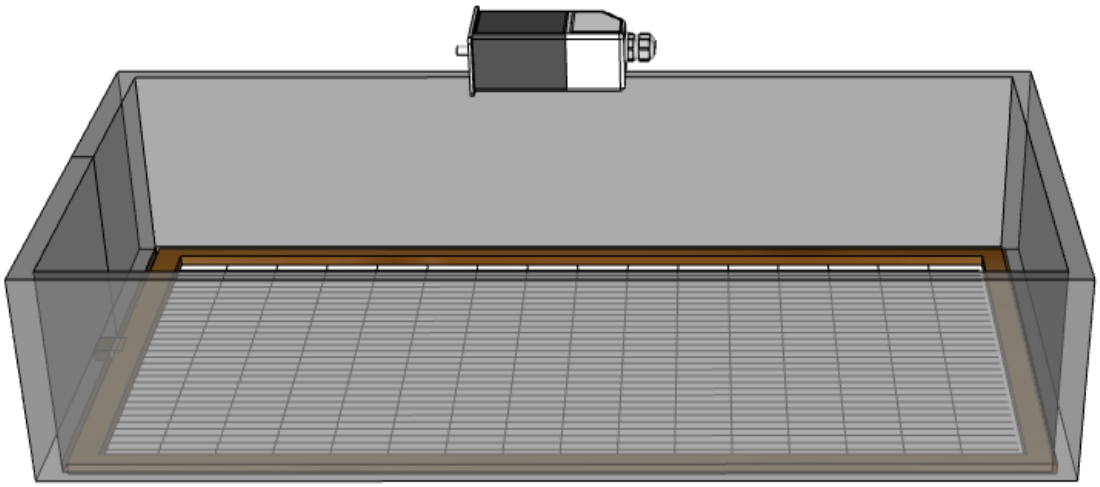


Figure 23. Hatching Container

The last level of the breeding system was the hatching container. This container housed the eggs that turned into larvae. It also had an ESP32 camera installed for monitoring and a load sensor to determine the plastic consumption and the population of the container. Transferring the larvae used a floor with an opening through a linear actuator and was transferred to the biodegradation container. This was done every 14 days or depended upon user discretion.

The measurements and dimensions of the measurement and dimensions of the hatching container were presented in table 19.

Table 19. Measurement and dimensions of the hatching container

Part	Length (in)	Width (in)	Height (in)
Inner part	15.3	7.7	3.1
Outer part	16	8.3	3.1

The mechanism used to open the hatching container was a pulley system wherein a DC motor was used for the opening and closing of the mesh screen. The circuit used a 2 relay module which helped reverse the motion of the pulley system. The figure below showed the circuit of the pulley system.

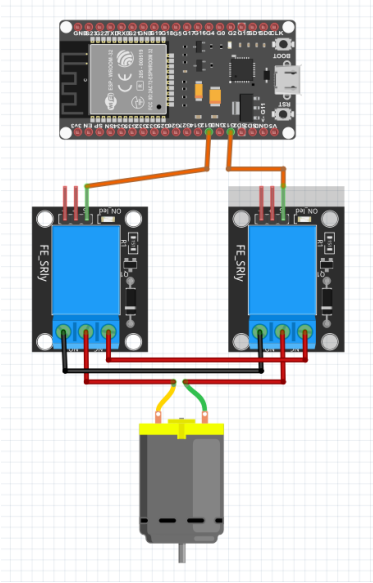


Figure 24. Circuit diagram of the pulley system

3.3.3 Design and Construction of the Biodegradation container and Frass bin

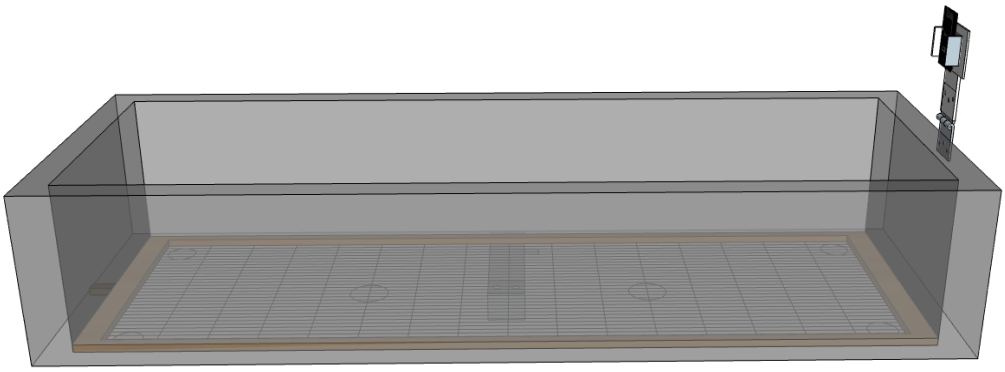


Figure 25. Biodegradation container

3.3.3.1 Biodegradation container

The biodegradation container housed the mature larvae and was the main population that had the biggest plastic consumption in the whole system. It had 1 Raspberry Pi camera, a load sensor, a vibrating motor, and a 1/30 in. mesh screen to segregate the frass. The frass was then transferred to the frass bin. This was done twice a month. To avoid overpopulation, the system had a population limit of 6000 superworms, and the excess was stored in other containers of the user.

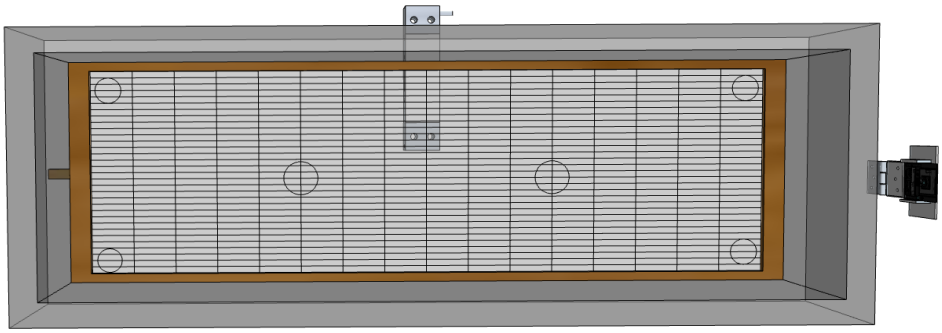


Figure 26. Placement of Load Sensor, ESP32 Camera and Vibration motor

The measurements and dimensions of the biodegradation container were presented in table 20.

Table 20. Measurement and dimensions of the biodegradation container

Part	Length (in)	Width (in)	Height (in)
Inner part	24.2	7.7	4.7
Outer part	25.9	9.3	4.7

3.3.3.2 Frass bin

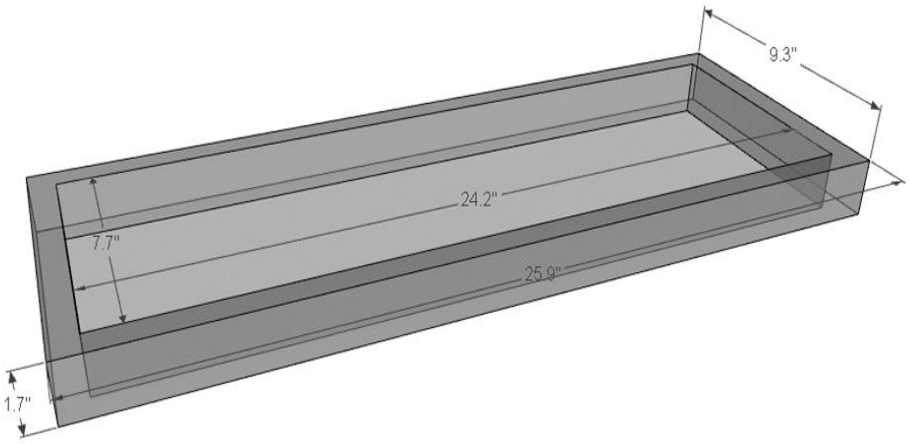


Figure 27. Frass bin

The frass bin alerted the user of the harvest time, and 300g was the weight threshold for the user to harvest the frass collected from the system. The measurements and dimensions of the frass bin were tabulated in Table 21.

Table 21. Measurement and dimensions of the frass bin

Part	Length (in)	Width (in)	Height (in)
Inner part	24.2	7.7	4.7
Outer part	25.9	9.3	4.7

3.3.4 Design and Construction of the Ventilation System

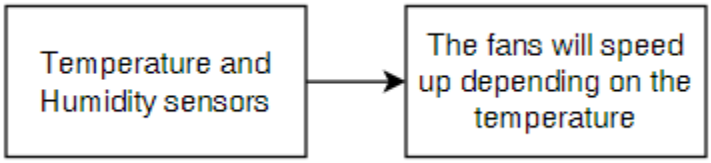


Figure 28. Ventilation system block diagram.

For the ventilation system, several sensors and mechanisms were required to create a livable ecosystem for Superworms. To achieve the highest hatchability of Zophobas Atratus (Superworm Beetle) eggs, the system maintained a closed temperature of 30°C. For the livable conditions of both Zophobas Atratus (Superworm Beetle) and Zophobas Morio (Superworm), several fans and a DHT11 Temperature and Humidity System were used.

3.3.4.1 Program and Design of the DHT11, RTC HW-084 Temperature Sensors and Fans

To program the DHT11 temperature sensor and the HW-084 temperature sensing functionality, the researchers first needed to install the DHT11 library and the DS3231 library in the Arduino software. Then, they configured the code to activate the fan if the threshold temperature was not reached. To test the code, they used the circuit diagram below.

Table 22. Sample data comparison between the thermal scanner and dht11.

	THERMAL SCANNER	DHT11
TEST NO. 1		
TEST NO. 2		
TEST NO. 3		

Since 2 temperature sensors were used in this system, the ventilation system was divided into 2 parts: the upper part of the enclosure, called the Mainboard, and the lower part of the enclosure, called the Container. Each temperature sensor was located in these areas and had a different set of thresholds. For the circuit diagram, refer to Figure 29.

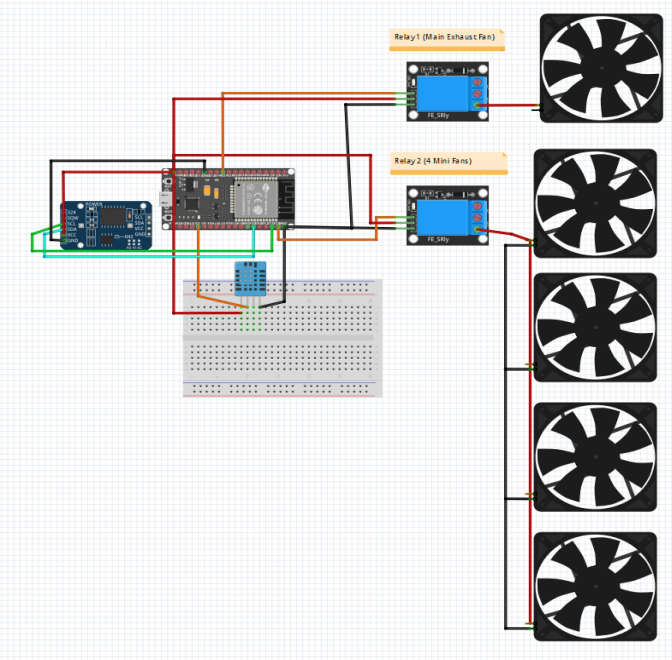


Figure 29. Temperature controlled fan circuit diagram

Figure 29 showed how the ventilation of the system worked. The fans were controlled via a mobile application. The system also had a manual

override for turning the fans on and off. The table below showed the temperature and its corresponding status of what happened to the fans at specific temperature values.

Table 23. Ventilation of the System

Temperature (°C)	Fan Status
25 - 27 °C	off
28 °C and above	on

The ventilation system consists of 6 120x25 mm fans, 4 for the intake placed on each side of the enclosure, 2 for the exhaust placed on the top of the enclosure. This is to ensure that the optimal airflow is achieved in the system. The dimensions and placement of the fans are shown in Figure 30.

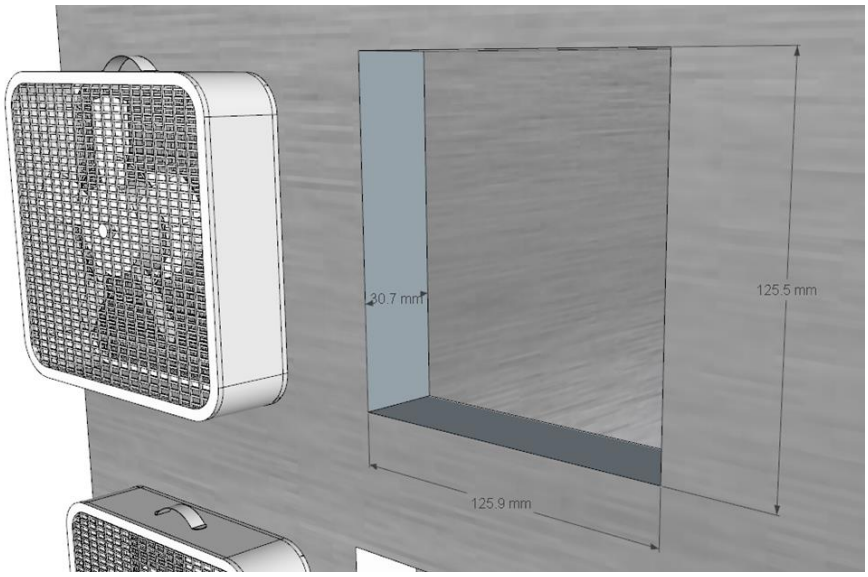


Figure 30. Dimensions of Fan and Placement Hole

Table 24. Fan and fan hole dimensions

Part	Length (mm)	Width (mm)	Height (mm)
Fan Hole	125.5	30.7	125.5
Fan	125	25	125

3.3.5 Design and Construction of the Skeletal and Enclosure

For the enclosure of the system, the researchers used a steel frame modeled as the holder of all the systems. The dimensions of the frame were first measured and constructed as shown in Figure 31.

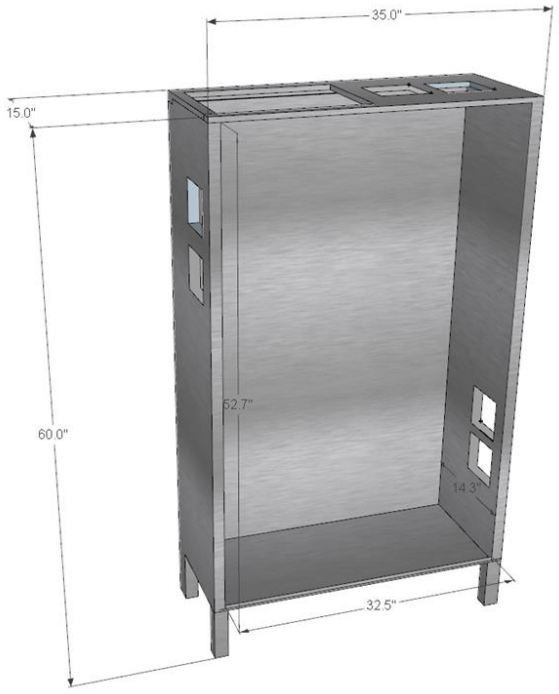


Figure 31. Enclosure of the system

As seen in Figure 31, the overall length of the design is 28 inches, the width is about 13 inches and the height is at 58 inches.

For the overall placement of each compartment, refer to Figure 32 and 33..

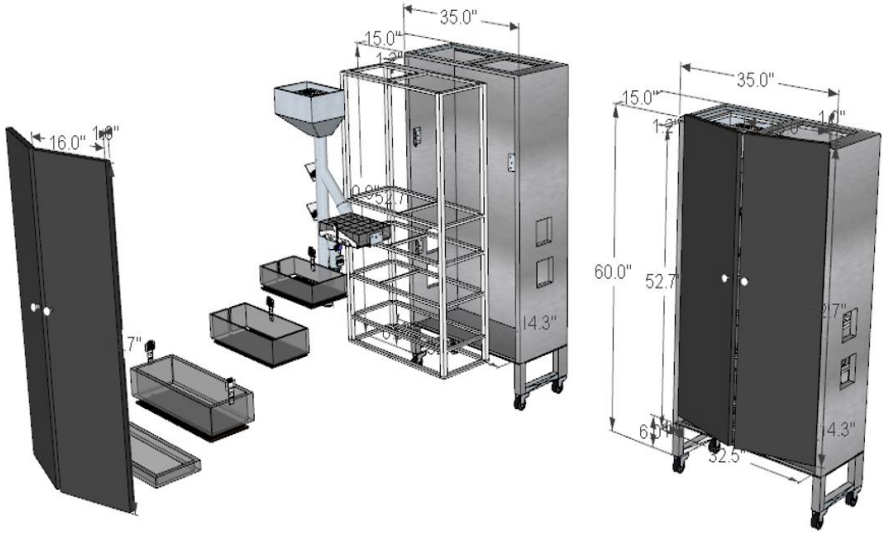


Figure 32. Exploded design of the system

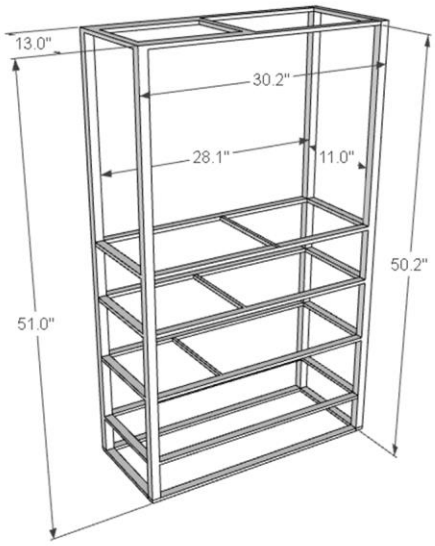


Figure 33. Skeletal Frame of the System

3.4 Development of Mobile Application for Monitoring and notification of the Breeding and Biodegradation System

The process of the transmission of data was illustrated in figure 34.

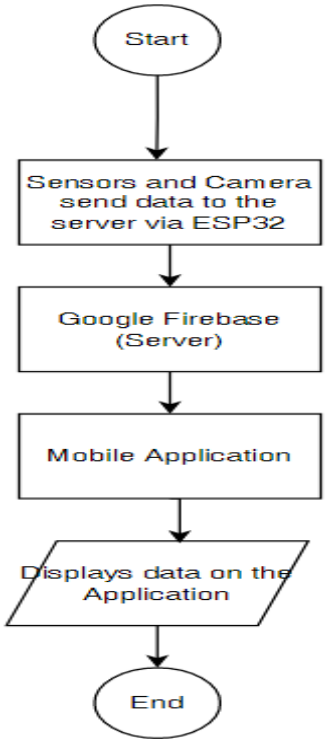


Figure 34. Transmission of data

3.4.1 Programming the Backend of the Mobile Application

3.4.1.1 Creating a Database using Google Firebase

Figure 35. Firebase Logo

Google Firebase offered two cloud-based database solutions that supported real-time data syncing. Realtime Database was Firebase's original database, providing an efficient solution for mobile applications that required synced states across clients in real-time. Cloud Firestore was Firebase's newest database solution for mobile application development, serving as the successor of the Realtime Database with a more intuitive data model. It also featured richer, faster queries and scaled further than its predecessor. In our case, Realtime Database was used since advanced querying, sorting, and transactions were not needed in the system. Realtime databases had extremely low latency, making them an ideal option for frequent state syncing.

3.4.1.2 Establishing the connection of the device to the database via ESP32

Connecting the ESP32 to the Realtime database required the installation of the “Firebase Arduino Client Library for ESP8266 and ESP32” in the Arduino IDE. The ESP32 must first be connected to the internet, including the network credentials in the code makes this possible. The database URL and firebase project API key, which are found when creating the database, are also included in establishing the connection between the ESP32 to the Realtime database. The device must sign in to the database, however the device must sign in to the database as an anonymous user so that it will not require a separate account for the ESP32. The user must enable this option when setting up the database.

3.4.1.3 Creating the Notification System

The Thunkable platform was used to connect it to the various sensors of the system and send the inputs of these sensors to the application. As discussed in subsection 3.4.1.2, the application obtained the inputs of the ESP32 from the Realtime database. This enabled the user to have access to the system whenever an internet connection was available. The expected notifications included the enumeration of the living conditions (temperature and humidity), the consumption rate of the system, the weight of the compartments, and alerts the user if the organisms and frass were ready to harvest.

3.4.1.4 Incorporating the Monitoring System to the Application

The application displayed the live stream of the state of the system using the installed cameras. Visual, block-based programming was used to display the status of the system in the application, serving as a reference for the user in monitoring the whole system.

3.4.1.5 Debugging and Testing of the program

The Debugging and Testing of the program were continuous during the time of the project-making to ensure that no bugs and malfunctions would happen in using the system. The program was evaluated as seen in the 3.7 Evaluation of User Acceptance Rating using the ISO standard 2506 section to ensure that the debugging and testing were executed properly.

3.4.2 Development of the User Interface of the Application

3.4.2.1 Designing of the Layout of the Application

The designing of the layout of the mobile application was done using Thunkable. Users could build native mobile apps using Thunkable's no-code platform without writing a single line of code. With the drag-and-drop interface, innovation could be easily unleashed and mobile-first solutions could be developed quickly. The initial step in creating the user interface for the application was to utilize Adobe Photoshop to construct a wireframe. Wireframes were used to build the basic structure of a page before visual design and content were added. After setting the foundation

for the user interface, Adobe Photoshop and Canva were used to design the various components, including the slides, buttons, and icons. After all the components and the layout were finished, Thunkable was utilized to build the application.

Figure 36. Logo of Thunkable, Adobe Photoshop, and Canva

3.4.2.2 Incorporating the backend functionalities to the user interface

To incorporate the backend to the user interface of the mobile application, the researchers fetched the logics and backend functionalities to the front end so that the information on the mobile application was updated with the data from the Realtime Database.

3.4.3 Importing the program into an apk file and User Testing

After debugging and simulating the application, the program was then imported into an APK file, which was installed on a device for user testing and debugging in the application. Further testing was done to resolve compatibility issues as well as optimize the overall performance of the mobile application.

3.5 Comparative Analysis of the Previous Versions of the Project

3.5.1 Defining and Implementing various Improvements for the Biodegradation of plastics

Researchers had found a different approach to effectively degrade different types of plastics. Instead of using mealworms, the project used superworms. From the previous research, the focus was mainly on degrading polystyrene, while this study had a larger scope. The study aimed to degrade different kinds of plastics such as polystyrene, expanded polystyrene, polyethylene, and polyurethane.

As for the software, the researchers used a mobile application for notification and monitoring and Google Firebase as a cloud database.

3.5.2 Testing Procedure for Data Gathering

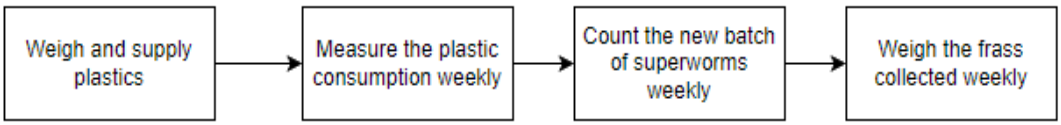


Figure 37. Block diagram of the testing procedure

The researchers gathered data on these parameters: Plastic (Polystyrene) biodegradation efficiency, population growth rate, and frass production and quality. To gather data on the Plastic biodegradation efficiency, an initial population of 4000 superworms served as the controlled group. These 4000 superworms were fed different plastics weekly. The order and amount of plastics supplied were tabulated in Table 14. To ensure that the superworms from the breeding system were not added, the breeding system kept only the new batch and notified the researchers if the population exceeded the size of the container. The researchers then kept the new

batch in another container for selling and replacement of the 4000 superworms in the biodegradation container.

The formula for determining the amount of plastic consumption was: Weight of plastic supplied (g) - the weight of biodegradation container (g) = Plastic Consumption

To gather data on the Population growth rate, the new batch of superworms was counted weekly to determine the number of superworms reproduced per week.

To gather data on frass production, the frass collected in the frass bin was weighed every week to determine the amount of frass produced by the 4000 superworms each week.

The new batches of superworms were added after gathering data for the adaptation of the automated breeding process for multi-plastic biodegradation.

Table 25. Testing procedure parameters

Period (weekly)	Plastic	Plastics Consumed (g)
Week 1	PS	
Week 2	PS	
Week 3	EPS	
Week 4	EPS	
Week 5	PE	
Week 6	PE	
Week 7	PU	
Week 8	PU	

3.5.3 Data Visualization of the Project

The time graph was developed using Google Sheets and Google Firebase. The Google Apps Script of Google Sheets was used to program the codes that were able to run and sync the data from the Firebase to Google Sheets. Every time the Firebase received new data, it automatically updated the Google Sheets and displayed it in a graph form. The graphed data in Thunkable was viewed using a web viewer, which linked the graphs from Google Sheets.

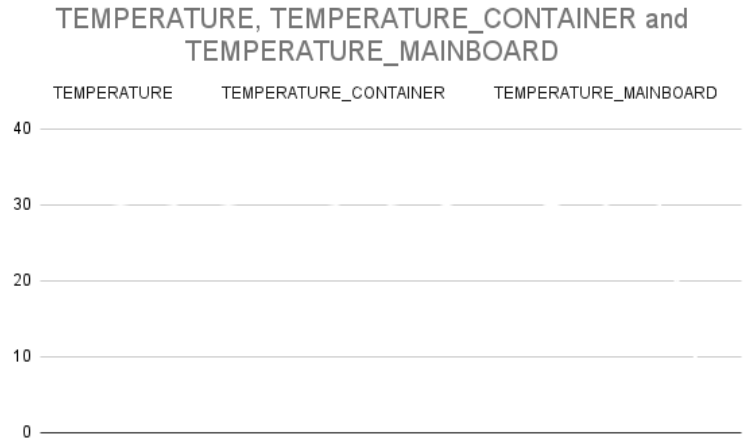


Figure 38. Time graph for temperature.

Figure 39. Time graph for weight.

Computational Model of Plastic Consumption of Different Time Frames:

Gathered Data (Weekly) (g):

Plastics

For Daily Consumption (g):

Plastics= Gathered Data (weekly)/7 days

To compute the daily consumption, we divided the weekly consumption to 7, which indicates the number of days within a week.

For Monthly Consumption (g):

Plastics= Daily Consumption x 30 days

To compute for the monthly consumption, we multiplied the computed daily consumption to 30, which indicates the number of days within a month.

For Yearly Consumption (g):

Plastics= Monthly Consumption x 12 months

To compute for the yearly consumption, we multiplied the computed monthly consumption to 12, which indicates the number of months within a year.

In 5 Years Consumption (g):

Plastics = Yearly Consumption x 5 years

To compute 5 years consumption, we multiplied the computed yearly consumption to 5 years.

3.5.4 Creating a Cross-Tabulation for the Previous Versions of the Project

The researchers used the Cross-tabulation method for the identification of the improved efficiency of the project entitled, "Automated Superworm Segregation System for Multi-Plastic Biodegradation using ESP32 with Computer Vision for Real-Time monitoring via Mobile Application", in the biodegradation of plastics and as well as the population growth rate in ensuring that the device was an ideal habitat for the organisms. This project was compared with its predecessors and used Plastic Biodegradation efficiency as its main parameter. The parameters considered were Plastic (Polystyrene) biodegradation efficiency, population growth rate, and frass production.

Table 26. Cross tabulation for previous versions of the project

Project Title	Polystyrene biodegradation efficiency	Population growth rate	Frass production
Automated Superworm Segregation System for Multi-Plastic Biodegradation using ESP32 with Computer Vision for Real-Time monitoring via Mobile Application			
Fully Automated Polystyrene Biodegrading System for Mealworm Segregation using Arduino Mega 2560 with ESP 32 for Wireless Notification			
Semi-Automated Biodegradation System for Polystyrene-Eating Mealworms with ESP8266 for Wireless Notification System			

3.6 Establishment of Business to Consumer Model for Marketability of Superworms and its Byproduct

3.6.1 Formulating the cost to profit ratio

For the formulation of the cost-to-profit ratio, a simple cost and return equation was used. The researchers had identified the main customers of the products of this system, namely, the superworms and their frass since it was observed that superworms and waxworms had a high fat and protein content compared to wax worms and crickets. Also, it was observed that out of all the species, superworms had vitamin D2 and were a good source of amino acids [29]. This was advertised to animal farms that used superworms as feed for their farms.

The superworm frass supported a higher proportion of rooting for the cacti than those cacti grown on oolong tea leaves. Superworm frass also resulted in an increase in average height of the cacti compared to oolong tea leaves [32]. Due to the large market of agriculture in the Philippines, the frass will be advertised to different farms and agricultural exchanges, because the frass of superworms are great fertilizers.

The observed prices from the various marketplaces and shops were:

For the superworms,

P100 for 200 medium-large sized superworms

For the frass,

P135 for 150 g of superworm frass.

3.6.2 Formulating the expected Return On Investment (ROI)

The researchers listed the costs for the components, tools, and equipment, and fabrication needed for the prototype, and then summed them to obtain all the costs for the project. Afterward, the researchers computed the total sale of superworms and their frass every month, which was subtracted from the overall cost of the project. The expected ROI of the project depended on the superworm and frass production of the system.

The Table 27 provides the total program cost, total program benefits, benefit-cost ratio, rate of investment and internal rate of return of the project as a whole.

Table 27. Sample of Simple Cost and Return Table

	Estimated Benefits					Resources needed in Units	Estimated Cost		
No. of Beneficiaries							No. Of Units	Unit Value	Total Cost
Direct Benefits	Year 1	Year 2	Year 3		Direct Cost				
1.					1.Utilities				
2.									
3.					Equipment, supplies, and materials				
4.					1. Printed materials				
					2. Furnishings and fabrication				
Indirect Benefits					3. Instructional Materials				
1.					4. Travel				

Total Program Benefits					Opportunity costs				
Benefit-cost ratio					1. Food				
Rate of Investment					2. Travel				
Internal Rate of Return (IRR)									

For the computation of costing:

Total Program Cost (C) =Labor(L) + Materials and Equipment

For Superworm sales:

Superworm Sales(S) = no. of superworms to be sold x selling price

For Return of Investment:

ROI = $c/c\,\frac{Sales - C}{C} x100\%$

3.7 Evaluation of User Acceptance Rating using the ISO standard 25063

3.7.1 Evaluate and Validate the Functionality of the Project using Interviews and Evaluation forms

The researchers gathered feedback regarding the performance of the project, Automated Superworm Segregation System for Multi-Plastic Biodegradation using ESP32 with Computer Vision for Real-Time monitoring via Mobile Application, and its ease of use, in order to determine its effectiveness and efficiency. The users were selected from the MRF employees of the selected site of deployment. A questionnaire was given to them and recorded for the continuous improvement of

the project. The evaluation questionnaire was based on the different factors of the ISO 25063, to determine the quality of the hardware and software, and a technical evaluation using a 5-point Likert scale.

Table 28. Evaluation questionnaire

Superworm Segregation with Monitoring System using Mobile Application					
Introduction: The researchers need to conduct an evaluation questionnaire from the users regarding the different factors and parameters in the functionality of the project entitled as, Automated Superworm Segregation System for Multi-Plastic Biodegradation using ESP32 with Computer Vision for Real-Time monitoring via Mobile Application					
Instructions: Please rate each of the parameters, and check one response per number based from these ratings; Check 1- if Strongly Disagree, 2- if Disagree, 3 - if Neutral, 4 - if Agree, and 5 - if Strongly Agree.					
Survey Statements	**Rating**				
	1	**2**	**3**	**4**	**5**
Functional Suitability					
1. The automation of the breeding process of the Superworms is more effective than the traditional method.					
2. The automated segregation of the Superworms is more efficient than the manual or traditional method.					
3. The live video clearly shows the organisms and ecosystem.					
4. The notification system for the updates on the life stages of the organisms are timely.					
5. The notification system for the harvest of frass in the application is effective.					
6. The information for the system's living conditions are well displayed in the interface of the mobile application.					

Usability					
7. Both the Hardware and Application are easy to use.					
8. The whole system needs minimum supervision during its entire operation.					
9. The design of the device is pleasing to the eye and accessible.					
Safety and Reliability					
10. The structure of the device is a safe habitat for the organisms.					
11. The construction of the device does not impose hazards to its users.					
12. The inlets, outlets, and wiring are intact and properly organized.					

3.8 Statistical Analysis

The research used a t-test as Statistical Analysis for a comparative analysis between the previous research, to determine the effectiveness of using the superworms in the biodegradation of plastics.

The t-test was a statistical analysis used when comparing the means of two groups or hypothesis testing to determine whether a hypothesis actually improved on the subject. The null hypothesis was the result of the usage of mealworms in the biodegradation of plastics, while the alternate hypothesis was the result of the usage of superworms in the biodegradation of plastics.

3.9 Project Work Plan

The proposed project work plan that the researchers will follow in performing this research is shown in Table 29:

Table 29. Project work plan

Research Activities	1	2	3	4	5	6	7	8	9	10	11	12
3.3 Development of an Automated Superworm Breeding and Segregation device through Integrated Sensors and Equipment												
3.3.1 Design and Construction of the Food Processing System												
3.3.1.1 Design and Construction of the Tubes												
3.3.1.2 Design and Construction of the Tube Switch Mechanism												
3.3.1.3 Design and Construction of the Stopper Mechanism												
3.3.1.4 Design and Construction of Weight Switch Mechanism												
3.3.2 Design and Construction of the Breeding and Segregation System												
3.3.2.1 Pupation Chamber												
3.3.2.2 Breeding Container												
3.3.2.3 Hatching Container												
3.3.3 Design and Construction of the Biodegradation system and Frass bin												
3.3.3.1 Biodegradation system												
3.3.3.2 Frass bin												
3.3.4 Design and Construction of the Ventilation System												
3.3.4.1 Program and Design of the DHT11, RTC HW-084 Temperature Sensors and Fans												
3.3.5 Design and Construction of the Skeletal and Enclosure												
3.4 Development of Mobile Application for Monitoring and notification of the Breeding and Biodegradation System												
3.4.1 Programming the Backend of the Mobile Application												

3.4.1.1 Creating a Database using Google Firebase												
3.4.1.2 Establishing the connection of the device to the database via ESP32												
3.4.1.3 Creating the Notification System												
3.4.1.4 Incorporating the Monitoring System to the Application												
3.4.1.5 Debugging and Testing of the program												
3.4.2 Development of the User Interface of the Application												
3.4.2.1 Designing of the Layout of the Application												
3.4.2.2 Incorporating the backend functionalities to the user interface												
3.4.3 Importing the program into an apk file and User Testing												
3.5 Comparative Analysis of the Previous Versions of the Project												
3.5.1 Defining and Implementing various Improvements for the Biodegradation of plastics												
3.5.2 Testing Procedure for Data Gathering												
3.5.3 Data Visualization of the Project												
3.5.4 Creating a Cross-Tabulation for the Previous Versions of the Project												
3.6 Establishment of Business to Consumer Model for Marketability of Superworms and its Byproduct												
3.6.1 Formulating the cost to profit ratio												
3.6.2 Formulating the expected Return On Investment (ROI)												
3.7 Evaluation of User Acceptance Rating using the ISO standard 25063												
3.7.1 Evaluate and Validate the Functionality of the Project using Interviews and Evaluation forms												

Chapter 4

Chapter 4 Results and Discussion

4.1 Project Technical Description

The project entitled SUPERB: Automated Superworm Segregation System for Multi-Plastic Biodegradation using ESP32 with Computer Vision for Real-Time monitoring via Mobile Application was an Internet of Things (IoT) based device that automated the regulation of the ideal environment of the Superworms. The superworms were used as feeders or the main component of the multi-plastic biodegradation feature of the system. The plastics that the superworms fed on were Polystyrene (PS), Expanded Polystyrene (EPS), Polyethylene (PE), and Polyurethane (PU) which were the recorded plastics that the Superworms could consume and digest. Within this biodegradation capability, the system introduced an automatic food distribution system that was composed of 3D printed mechanisms, Servo Motors, and Load sensors. These mechanisms aided in redirecting the distribution of plastics in each container.

After the distribution system, the system also used an automated ventilation system that used DHT11 as a temperature sensor, 4 side fans, and 1 exhaust ceiling fan to make sure that the system was cooled down to its ideal temperature, which was 27 C. Making the temperature as ideal as it could be helped increase the production of superworms.

With the production of superworms taking place, the automated breeding and segregation system helped make the life cycle of superworms easier. The first container housed 25 larvae that turned into beetles after 2-4 weeks. The system automatically

transferred these beetles after 4 weeks by opening the flooring of the container using a Linear actuator. In the 2nd container, which was the breeding container, this was the container that allowed the beetles to breed and produce eggs. After 2 weeks, the system used the attached vibration motors in the mesh screen flooring of the container. This transferred the eggs to the 3rd container or the hatching container. This hatching container housed the eggs, and after 2 weeks, there were visible superworms, and these superworms were transferred to the 4th container, the biodegradation container, using the pulley system with a DC motor. This opened the flooring and transferred all of the larvae to the next container. The biodegradation container was the container that fed on the majority of the plastics that the system supplied. To ensure that the container was a clean container, there were 6 vibration motors attached to its mesh screen flooring. This removed the frass of the superworms and transferred it to the last container, the frass container. This frass container notified the user if the attached load cell sensed that the container had 300 grams of frass. This signaled the user to harvest it.

And to keep track and ensure that the system provided the ideal environment, it had a computer vision that used YOLOv4 to help count the organisms present in the system. This proved that the system helped increase the production of the superworms that helped in biodegradation and also made superworm breeders more profitable. All of this data was shown in the developed mobile application using Thunkable and was also recorded using a time graph. To ensure that the system connected to the internet, it used an ESP32 and directly connected to the cloud database, Google Firebase.

All of these features and functions were mainly responsible for making the system as efficient as possible.

4.2 Project Structural Organization

4.2.2 Design and Construction of the Food Processing System

The system started with the distribution of plastic which served as the primary food source for the superworms.

4.2.2.1 Design and Construction of the Tubes

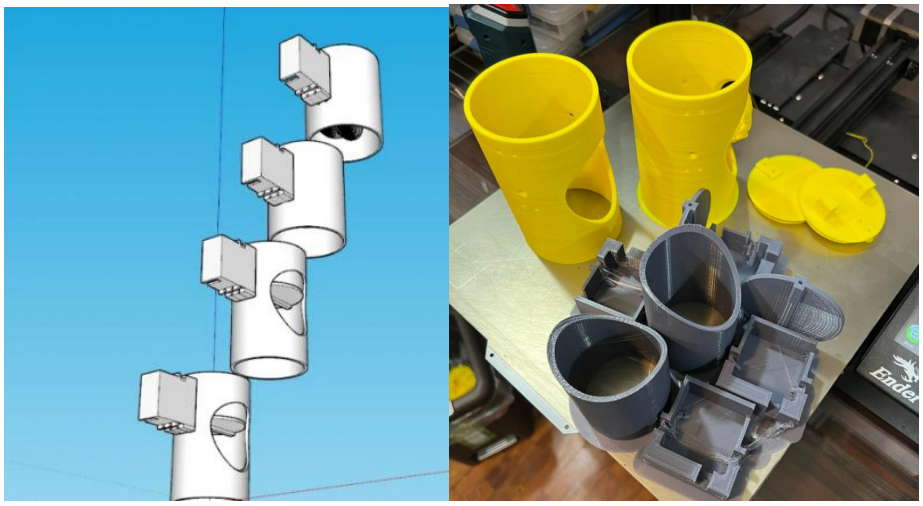


Figure 40. Design and Construction of the Tubes.

4.2.2.2 Design and Construction of the Tube Switch Mechanism

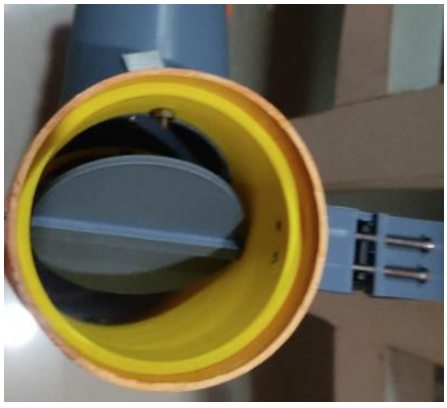


Figure 41. Tube Switch Mechanism.

4.2.2.3 Design and Construction of the Stopper Mechanism

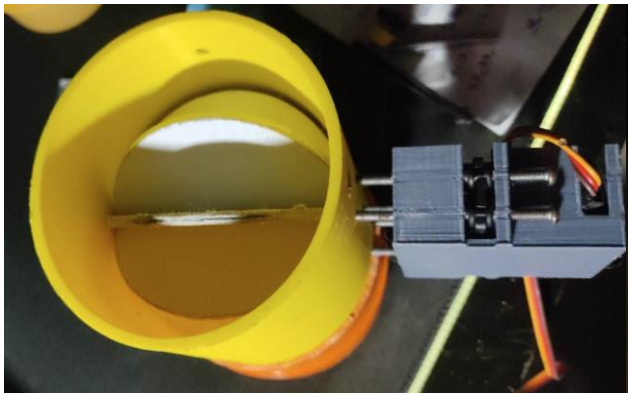


Figure 42. Stopper Mechanism

4.2.2.4 Design and Construction of Weight Switch Mechanism

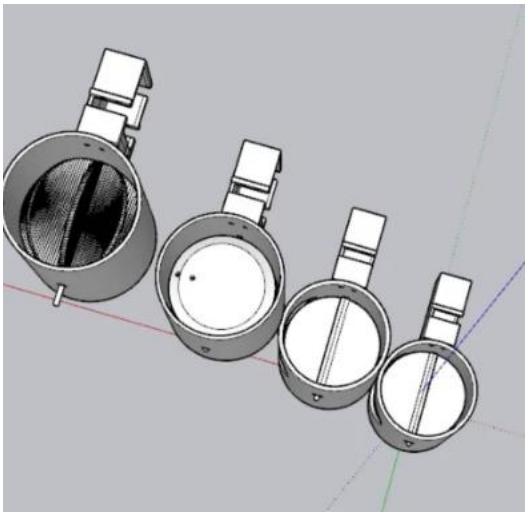


Figure 43. Weight Switch Mechanism.

Load Sensor

The load cell was used to measure the weight of the superworms and plastics. The goal was to determine the amount of plastic being consumed by the superworms, which provided insight into the efficacy of the project.

Servo Motor

The servo motors worked by redirecting the plastics through the use of sliced plates located in between the tubes. The load cell was used to measure the weight of the plastics, and the servo motors were then triggered based on that measurement. This allowed for a controlled and automatic redirection process that was based on the weight readings obtained from the load cell.

4.2.3 Design and Construction of the Breeding and Segregation System

Automatic Segregation System

The system used different automated segregation or rearing mechanisms to separate the different life stages of a Superworm.

4.2.3.1 Pupation Chamber

Container 1. Pupation Chamber with Linear Actuator

Figure 44. Pupation Chamber

Figure 44, the Pupation Container, housed 25 superworms/larvae that were isolated from the rest of the population to become beetles. Since the project aimed to become a sustainable habitat and system for the Superworms, the isolation within the Pupation chambers helped establish the first stage of their life cycle. The pupation of the worms lasted around 2 weeks, and then the organisms entered the pupa stage. After another 2 weeks, the pupae transformed into beetles, which were the finished product of Container 1.

The trigger of the Linear Actuator to move the beetles to the 2nd Container was time-based. The system used an RTC module and opened the chambers after 4 weeks of larvae placement. The activation of the Linear Actuator lasted for 1 minute and utilized an input voltage of 5V.

4.2.3.2 Breeding Container

Container 2. Breeding Container with Vibration Motors

Figure 45. Breeding Container

Figure 45, container 2 served as the breeding container where the beetles bred and produced eggs. These eggs were transferred to the 3rd container using the

5 flat vibration motors attached to the mesh screen of the container. This process was also time-based, using the RTC module as a reference. The vibration motors activated after 2 weeks of the initial transfer of beetles. The flat vibration motors used an input voltage of 3.5V and ran for 1 minute to ensure the segregation of the eggs and beetles. During the testing phase of the project, a load sensor was also incorporated to measure the plastic consumption of the beetles.

4.2.3.3 Hatching Container

Figure 46. Hatching Container with Pulley System

Figure 46, container 3 served as the Hatching container where the eggs grew into Superworms. This container utilized an attached DC motor that functioned as a pulley system-like mechanism. This mechanism opened and closed the flooring of the 3rd container to transfer the grown superworms into the Biodegradation Container.

The pulley system was also time-based, activating after 4 weeks. It opened the flooring to transfer the grown worms into the biodegradation container. The activation duration was 30 seconds to facilitate the transfer of the organisms. During the testing phase of the project, a load sensor was incorporated to measure the plastic consumption of the small growing superworms.

4.2.4 Design and Construction of the Biodegradation container and Frass bin

4.2.4.1 Biodegradation system

Container 4. Biodegradation Container with Vibration Motors

Figure 47. Biodegradation Container

Figure 48. Vibration Motors attached to the Biodegradation Container

Figure 47 and 48, container 4 served as the Biodegradation container, which was the main container where the majority of the population live. The initial population used in the project was 4000 superworms, and the container had three different attachments: 6 flat vibration motors, a Raspberry Pi camera, and a load sensor. Each of these attachments had different functions and uses for this container.

The container had 6 flat vibration motors that were used to separate the frass or byproduct of the superworms, which was then transferred to the 5th container. This mechanism was time-based, activating every 24 hours with a duration of 1 minute.

A webcam was installed in the container to aid in monitoring the organisms and count them using the YOLOv4 machine learning algorithm. This helped

identify if the system provided the ideal environment and habitat for the Superworms.

The load sensor continuously measured the weights of the superworms and plastics. This information was crucial in determining the overall plastic consumption by the superworms. The load sensor and computer vision system functioned continuously, allowing for the production of a time graph that visualized the data and demonstrated the efficacy of the project.

4.2.4.2 Frass bin

The Frass bin served as the final container that contained the frass of the superworms. It had a load sensor attached below to measure and record the weight of the frass produced. When the load sensor detected that the frass reached 300 grams, it alerted the user through the mobile phone, prompting them to harvest the frass out of the container.

Figure 49. Frass Bin

4.2.5 Design and Construction of the Ventilation System

4.2.5.1 Program and Design of the DHT11, RTC HW-084 Temperature Sensors and Fans

Temperature-Controlled Ventilation System

The system utilized two different areas for temperature reading and control. The first area was for the Main board temperature, which served as a trigger for the upper exhaust fan. When the Main board temperature reached a certain threshold, it activated the upper exhaust fan to help cool down the system.

The second area was for the lower containers' temperature, which served as a trigger for the 4 side fans. When the temperature of the lower containers exceeded a specific level, it triggered the activation of the 4 side fans.

Upper Exhaust Fan

The upper exhaust fan was installed in the top right corner of the system, directly above the main board. Its placement allowed for quick and efficient regulation of the system's temperature. The fan was supplied with a 12V input voltage using the switching power supply integrated into the main board.

Once the temperature of the main board reached 28°C, the fan automatically turned on, effectively dissipating the heat. It remained operational until the temperature dropped below 27°C, at which point the fan automatically turned off, ensuring that the temperature remained within the desired range.

Figure 50. Upper Exhaust Fan.

Side Fans

The 4 12V, 120mmx120mm fans were installed on the sides of the system, specifically in the biodegradation container where the majority of the population resided. These fans were strategically positioned to provide efficient cooling to the container.

Similar to the upper exhaust fan, these fans were designed to automatically turn on when the temperature of the container rose to 28°C.

Their activation helped regulate the temperature by circulating air and dissipating heat. Once the temperature dropped below 27°C, the fans automatically turned off to maintain the desired temperature range within the container.

.

Figure 51. Side Fans

4.2.6 Design and Construction of the Skeletal and Enclosure

Figure 52. Enclosure.

Control Panel

The system incorporated a control panel equipped with dedicated physical buttons, allowing for manual control and override of the system in case of malfunction or specific requirements. An I2C 16x2 LCD display was utilized to provide a visual interface, organized into a directory based on the different functions and components of the prototype.

The control panel's physical buttons enabled users to navigate through the directory and select specific functions or system parameters. This provided a user-friendly interface for monitoring and managing the system's operations, allowing for manual intervention when necessary.

The directory is as follows:

1. Ventilation
    1.1 Temperature
        1.1.1 Main Board Temp
        1.1.2 Container Temp
    1.2 Exhaust Fan

1.4 Container Fan

2. Weights
    2.1 Biodeg Container

    2.2 Frass

3. Segregation

    3.1 Linear Actuator

    3.2 Pulley

    3.3 Vibration 1

    3.4 Vibration 2

4. Food Dispenser

    4.1 Calibrate

    4.2 Manual Control

        4.2.1 1st Part

        4.2.2 2nd Part

        4.2.3 3rd Part

        4.2.4 4th Part

5. Time and Date

    5.1 Update Time

6. Shutdown
    6.1 On/Off

This was the directory that was displayed in the LCD attached outside of the prototype.

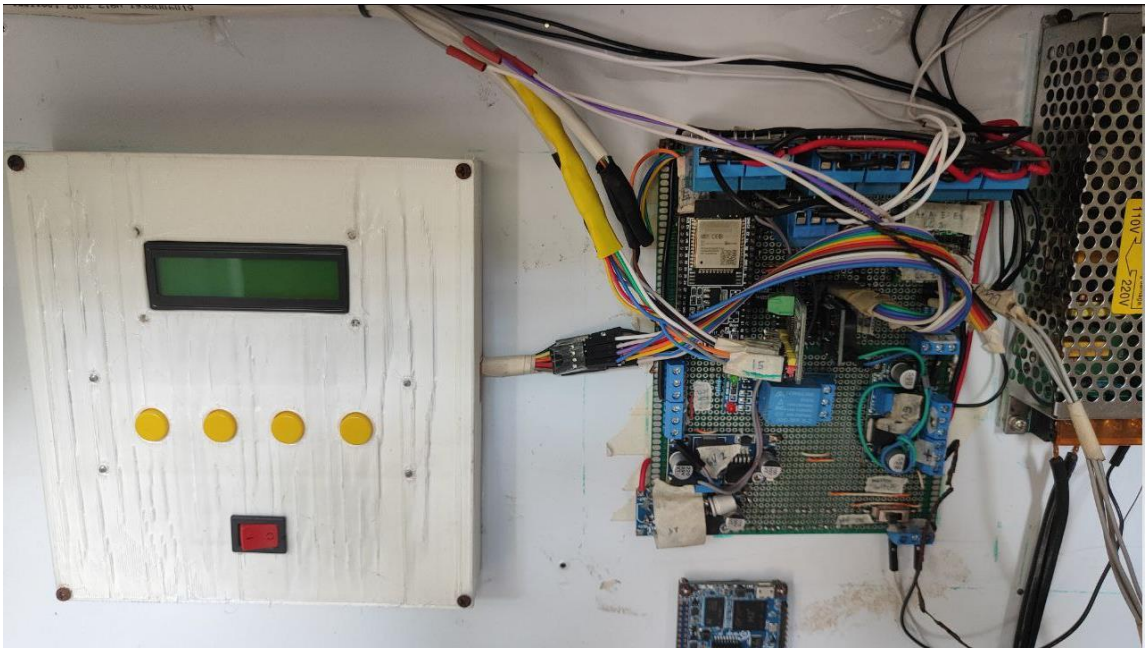


Figure 53. Control Panel.

4.2.7 Development of Mobile Application for Monitoring and notification of the Breeding and Biodegradation System

The mobile application displayed and provided access to all the data measured and recorded by the system. It included information such as temperature, humidity, container weights, manual controls, time graph, and computer vision data. Users were able to observe and control these variables through the application.

4.2.7.1 Programming the Backend of the Mobile Application

4.2.7.1.1 Creating a Database using Google Firebase

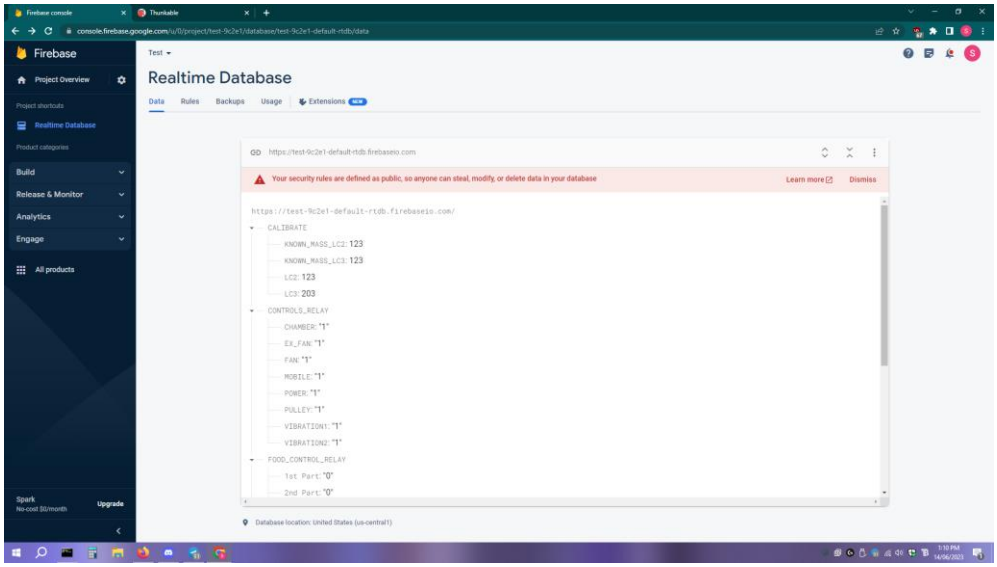


Figure 54. Realtime Database

4.2.7.1.2 Establishing the connection of the device to the database via ESP32

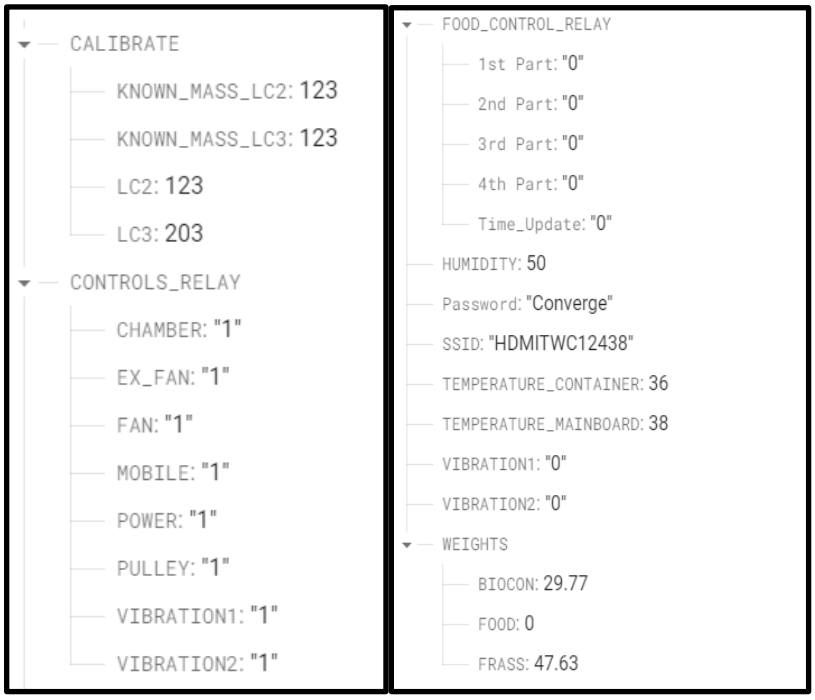


Figure 55. Database fields

Figure 55 shows the connection and real-time transferring of data from the system to the Google Firebase as the system's main database.

4.2.7.1.3 Creating the Notification System

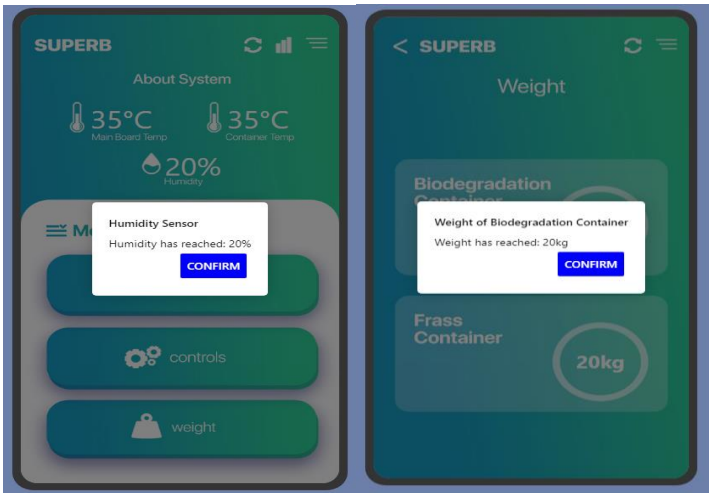


Figure 56. Weight and Temperature Notification.

The notification system, developed using the Thunkable platform, was seamlessly connected to the sensors within the system. It facilitated the transmission of sensor inputs to the mobile application in real-time. By leveraging a real-time database, the ESP32 relayed information regarding the weight and temperature measurements to the application. This enabled users to receive timely updates and access accurate information about the system's status via notifications.

4.2.7.1.4 Incorporating the Monitoring System to the Application

4.2.7.1.4.1 Camera livestream

Camera Livestream

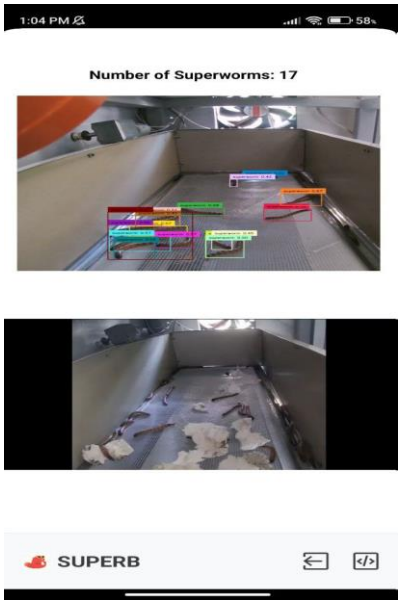


Figure 57. Livestream

Figure 57 shows the connection of the ESP32cam and its livestream that is being displayed in our application

4.2.7.2 Development of the User Interface of the Application

Figures 58 and 59 shows the user interface of the mobile application and its backend programming using the Thunkable platform:

4.2.7.2.1 Designing of the Layout of the Application

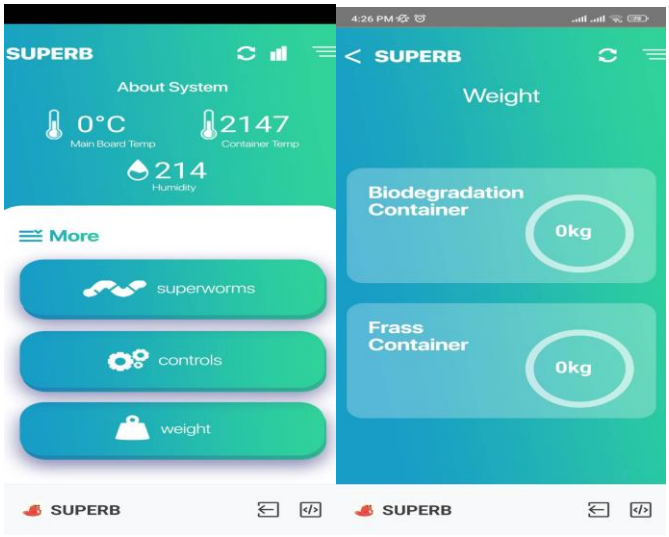


Figure 58. Display of data in the mobile application.

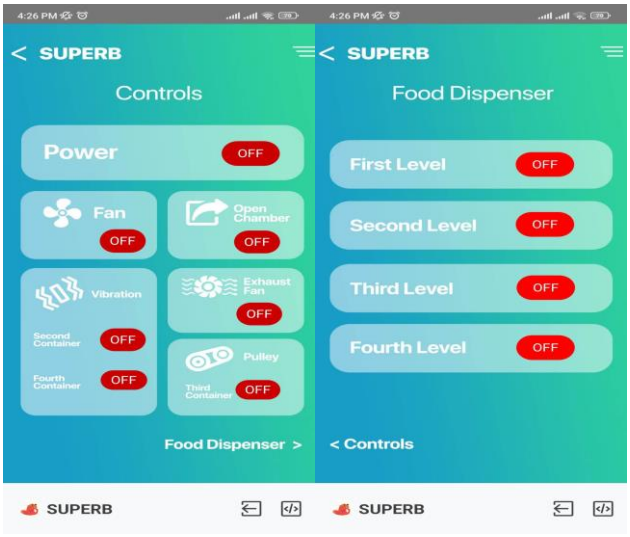


Figure 59. Manual controls buttons in the mobile application.

4.2.7.2.2 Incorporating the backend functionalities to the user interface

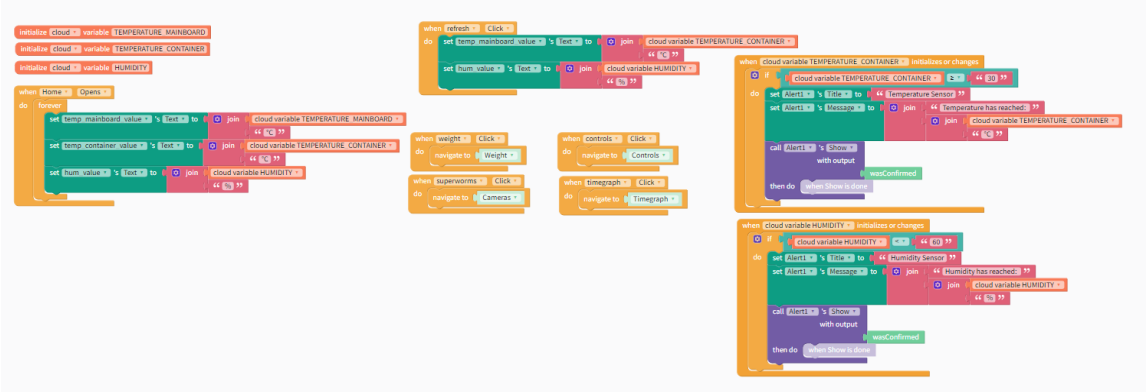


Figure 60. Backend of homepage

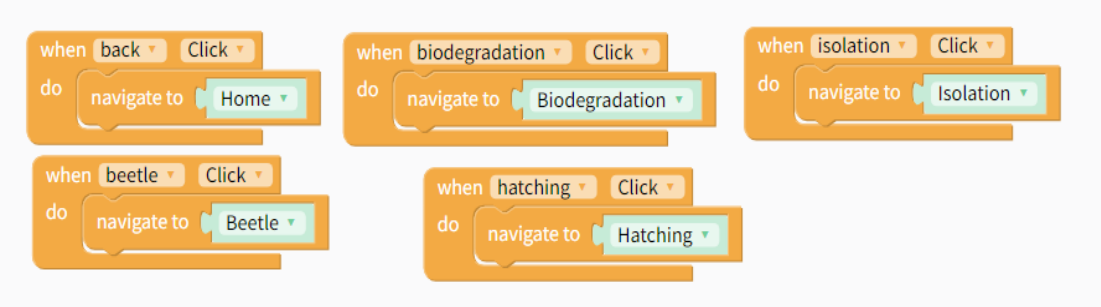


Figure 61. backend of camera selection page.

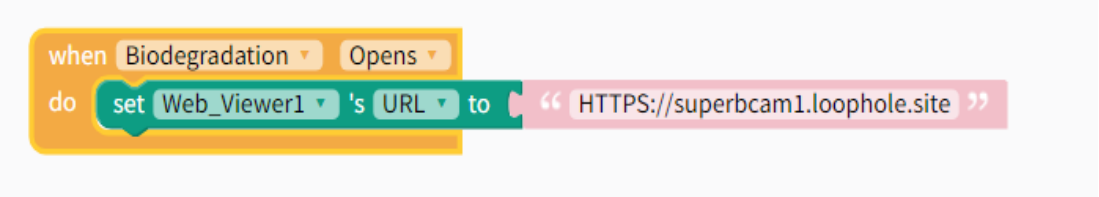


Figure 62. backend of camera page.

Figure 63. Backend of control page.

Figure 64. Back of food dispenser control page.

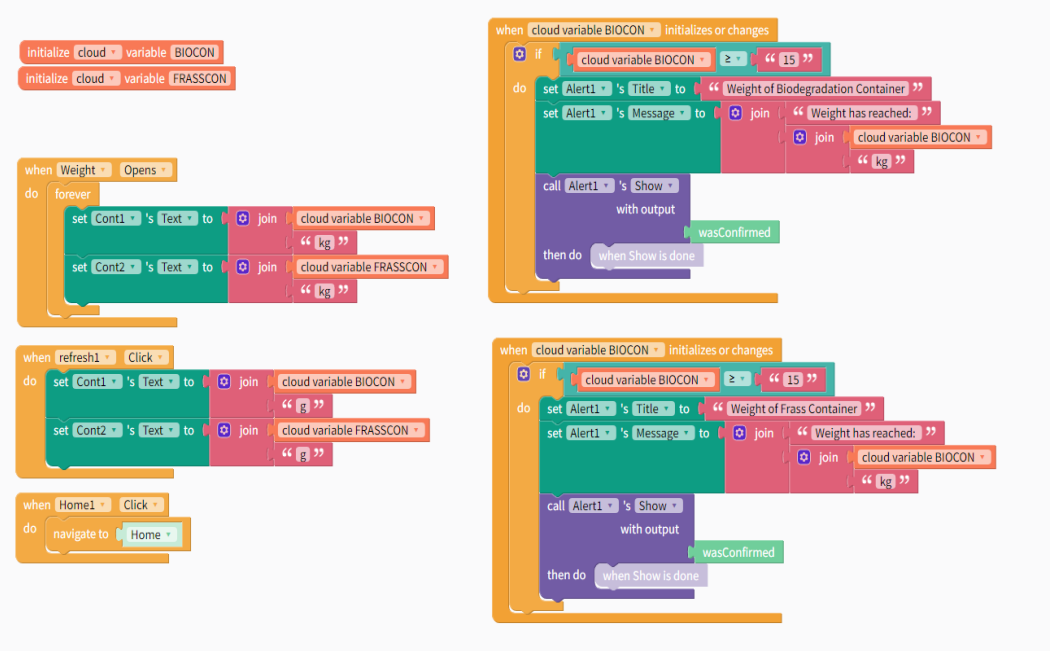


Figure 65. backend to display and notify the weight.

```
function myFunction() {
  var database_ref = "/";
  var url = "https://test-9c2e1-default-rtdb.firebaseio.com" + database_ref + ".json";

  var response = UrlFetchApp.fetch(url);
  var data = JSON.parse(response.getContentText());

  var sheetName = "timegraph";
  var sheet = SpreadsheetApp.getActive().getSheetByName(sheetName);
  if (!sheet) {
    sheet = SpreadsheetApp.getActive().insertSheet(sheetName);
  }

  var headers = [];
  for (var key in data) {
    headers.push(key);
  }

  var values = [];
  for (var key in data) {
    values.push(data[key]);
  }

  sheet.appendRow(values);
}

```

Figure 66. Backend of time graph

Object Detection and Counter

YOLOv4 was utilized for object detection and counting purposes due to its exceptional speed and accuracy in identifying objects within images or videos. This state-of-the-art deep learning model proved to be particularly well-suited for real-time applications, as it could rapidly process visual data and provide accurate results. The utilization of YOLOv4 ensured a high level of precision and efficiency in object detection and counting tasks, contributing to the overall effectiveness of the system.

YOLOv4 object detection and counter

The YOLOv4 algorithm was employed to analyze the images obtained from the webcam. This integration facilitated the automatic detection and counting of the Superworms present in the images. By leveraging the power of YOLOv4, the monitoring and counting process became significantly more efficient and accurate. The algorithm's advanced capabilities provided valuable insights into the Superworms' habitat and environment, aiding in the assessment of their well-being and contributing to the overall success of the system.

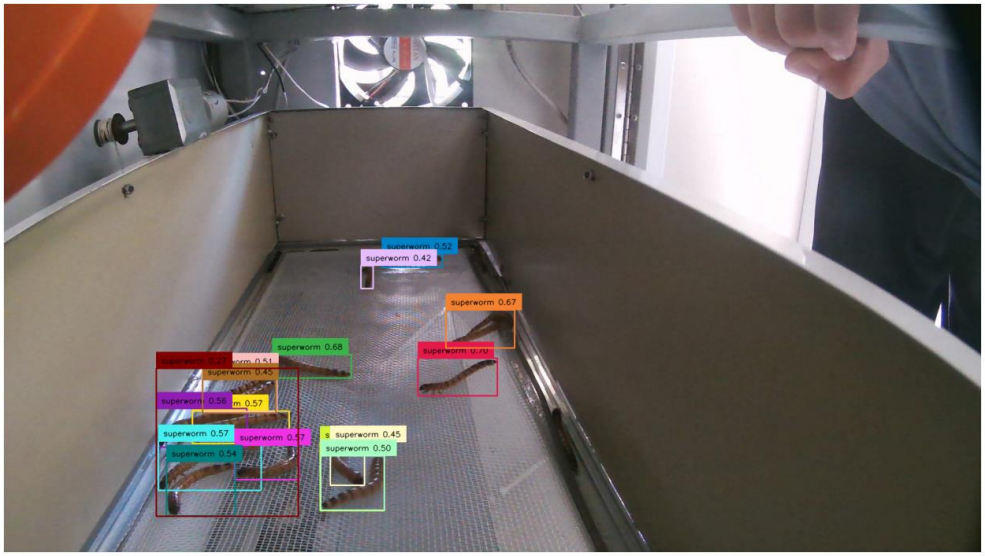


Figure 67. Trial of Superworm Detection and Counter

Table 30. Trial of Superworm Detection and Counter

Epoch	Accuracy (%)	Confidence rate (%)
1	43.5	42.1
2	48.2	46.8
3	51.7	50.3
4	56.1	54.9
5	59.8	58.6
6	62.4	61.2
7	66.3	65.1
8	69.7	68.5
9	72.2	71.0

4.2.8 Data Visualization of the Project

The project recorded in an hourly increment and showed the time graph presented in the mobile application.

Time Graph

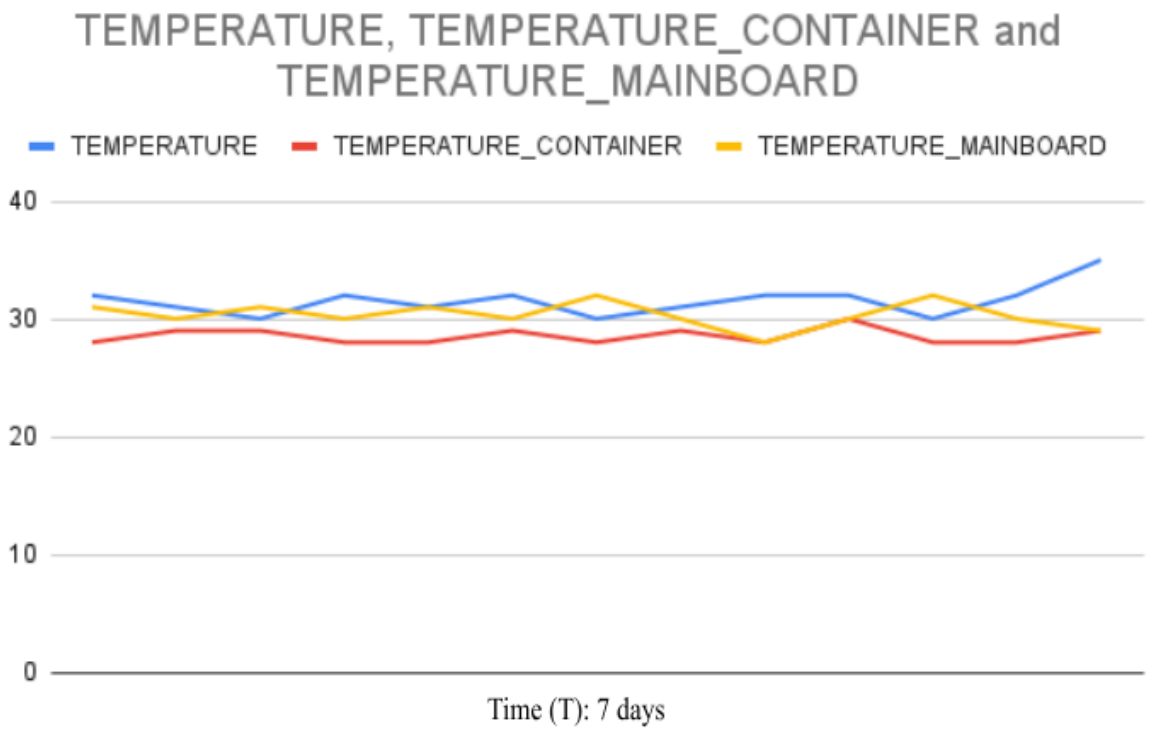


Figure 68. Time graph in the mobile application.

4.3 Project Limitation and Capabilities

The project was composed of different sensors and microcontrollers that were integrated as a whole system. The different systems were mainly the biodegradation system, breeding and segregation, food distribution system, and ventilation system.

The biodegradation system was populated with 4000 superworms, which served as the main decomposers of the various plastics used in the project. This population size was considered ideal for the testing phase to prevent overpopulation and overheating of the organisms, taking into account the size of the container and the project itself. However, this limitation meant that the project could not handle large amounts of plastics, such as kilos of plastics in a week. The scalability of the project was designed to address this issue, allowing for easy scaling and increased plastic consumption by adding more biodegradation containers.

The project focused specifically on five different types of plastics: Polystyrene (PS), Expanded Polystyrene (EPS), Polyethylene (PE), and Polyurethane (PU).These plastics were selected based on extensive studies and research that confirmed the Superworms' ability to consume and biodegrade them effectively.

The breeding and segregation system utilized various mechanisms, including a linear actuator, vibration motors, and a DC motor operating as a pulley system. These mechanisms were specifically designed and calculated for this particular system, taking into account the predetermined weight and number of organisms in each container. In the event that the containers' dimensions were to change, the mechanisms would need to be adjusted accordingly to accommodate the new container sizes.

The food distribution system was equipped with 4 servo motors responsible for redirecting the plastics using sliced plates located between the tubes. These redirections were triggered based on the weight measurements obtained from the load sensor. The flow of food followed a specific process: pre-processed small-sized plastics were placed in the

prototype's funnel, where they underwent regulated dropping or flowed through the food regulator mechanism within the tube. The plastics were then measured by the load cell inside the tube. Depending on the predetermined weight threshold for each container, the servo motors with attached slice plates would redirect the flow of plastics to the respective containers. This cycle occurred on a weekly basis, with different plastics inserted into the funnel each week. It's important to note that the load sensor had a weight limit of 5kg, and anything above this limit would not be accurately measured by the sensor.

The ventilation system consisted of four 120x120mm fans installed on each side of the container, along with an 8-inch diameter exhaust fan positioned at the top of the prototype. These fans were connected to a DHT11 temperature sensor and an RTC module, which also included a built-in temperature sensor. The DHT11 temperature sensor was strategically placed near the biodegradation container to monitor the temperature. Based on the measured temperature, the side fans were controlled accordingly. When the temperature reached the threshold of 28°C, the side fans would turn on, and they would turn off once the temperature dropped below 27°C. Similarly, the upper exhaust fan operated based on the temperature readings from the RTC module. It followed the same temperature threshold, activating when the temperature exceeded 28°C and deactivating when it fell below 27°C. However, it's important to note that this system was unable to increase the temperature of the environment in very cold locations.

To enhance user control and interaction with the system, the researchers included a control panel within the mobile application as well as physical buttons. The physical control panel utilized a 16x2 i2c LCD display, accompanied by four buttons for navigation and input. However, it's important to note that the limitation of this display was its ability

to show only 16x2 characters, which could appear small and challenging to read for individuals with visual impairments or eye problems.

On the other hand, the controls provided in the mobile application offer a responsive user interface with a latency of approximately 1 second. It's important to consider that the latency may vary depending on factors such as the strength and speed of the internet connection. Lower signal strength or slower internet speeds can potentially increase the latency, affecting the responsiveness of the controls within the mobile application.

4.4 Project Evaluation

For the initial testing of the project, the researchers have compared the weighing scale and load cell sensor to see if the load cell provides an accurate measurement.

Table 31. Initial testing (Weighing Scale vs. Load Cell)

Test No.	Item	Weighing Scale (g)	Load Cell (g)
1	Hammer	700	713
2	Metal Bar	200	204
3	Pliers	250	252

Table 32. Organism Population (Weekly)

Week	Worms	Pupa	Beetle
1	4000	0	0
2	3885	0	0
3	3885	15	0
4	3885	15	0
5	3885	3	7
6	3885	1	9
7	3885	0	10
8	3885	0	10

In the Table 32, 15 Superworms were taken from the initial 4000 worm population for it to pupate. After 2 weeks, the 15 isolated Superworms became pupae. During the 2 weeks of pupation period, 5 of the pupae died, while 10 became beetles. In this period, the researchers continued to manually count the Superworms' population.

Figure 69. Manual Counting of Worm Population

Figure 70. Pupation Chamber

The Superworms were separated within the pupation chamber to enter the pupation stage. After a specific period of time, the linear actuator is activated to lower the flooring of the chamber, allowing the beetles to be transferred into the breeding container.

Figure 71. Beetle's Breeding Container

The beetles mate inside the breeding container to produce the eggs needed for hatching newly bred superworms. After a specific mating period, the vibration motors

attached to the mesh screen were activated to sift the eggs and transfer them to the hatching container.

Figure 72. Hatching Container and Biodegradation Container

In the hatching container, the eggs from the previous container hatched. Each hatched Superworm stayed in the hatching container for a specific period of time. Once the hatching period was over, the pulley system was activated to lower the mesh screen flooring. This allowed the hatched Superworms to drop into the biodegradation container, where they consumed a significant amount of plastics. Vibration motors attached under the mesh screen of the biodegradation container ensured that the Superworm frass was sifted and transferred into the frass container. But since there is no eggs produced, micro-sized worms were used to see if the container can hold smaller sized worms.

Figure 73. Frass Container

The frass from each container was transferred to the frass bin. After a specific period of time had elapsed and the weight of the frass container was assessed using weight sensors, the user would receive a notification indicating whether it was permissible to remove the frass from the container.

For the temperature sensor, the researchers have used a thermometer to measure the temperature and will be compared to the DHT11 of the prototype, the data recorded is shown in the table below.

For the automation and manual controls of the mechanisms, it was successfully developed and tested as it can function for its purpose. From the automated distribution system up to the segregation system. However, in the reproduction of the superworms, the project did not produce new offsprings due to biological circumstances. From this, the researchers' goal is testing the effectiveness in segregating the organisms by buying smaller worms, to test its segregation capability.

In the aspect of the biodegradation capability of the system, it showed successful results as it had biodegraded 5.7 g of PS and EPS plastics daily, followed by the PU with a value of 2.22g and PE with 0.46g. With a forecasted 171 g of styrofoam per month, 66.6 of PU and 13.8 g of PE in a month.

In regards with the marketability of the Superworms, the researchers have interviewed breeders and users to ensure that the data is equal to the present day. The Superworm breeder and seller named Ed Peruelo, stated that breeding superworms is a great business opportunity since it has a high demand, from pet owners to pet shop owners. Table 33 and 34 is the questionnaire that the researchers used for the market research.

Figures 74, 75, 76, 77, 78, and 79 are the documentation for all the objectives of the study.

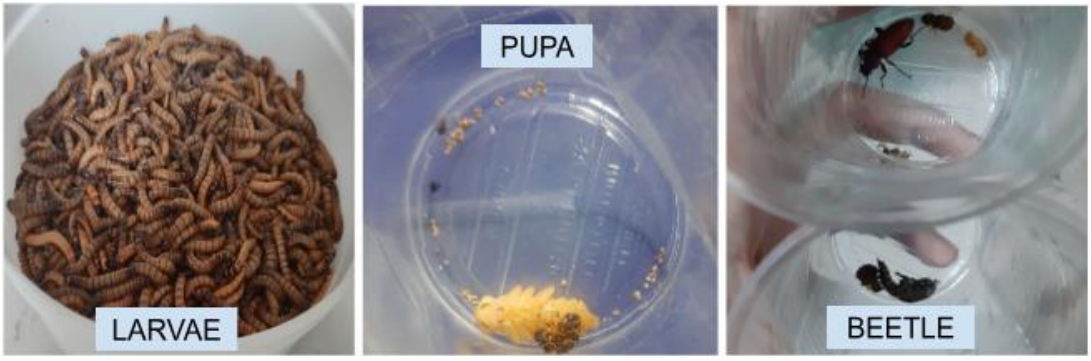


Figure 74. Life Cycle of Superworms.

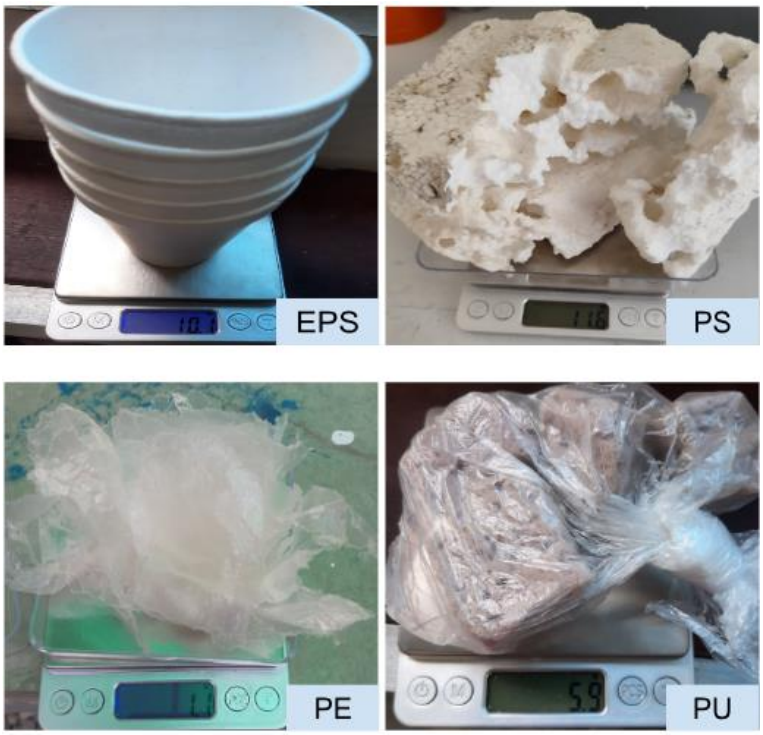


Figure 75. Plastics left after 1 week of Data Gathering.

Figure 76. Plastics left after 2 weeks of Data Gathering.

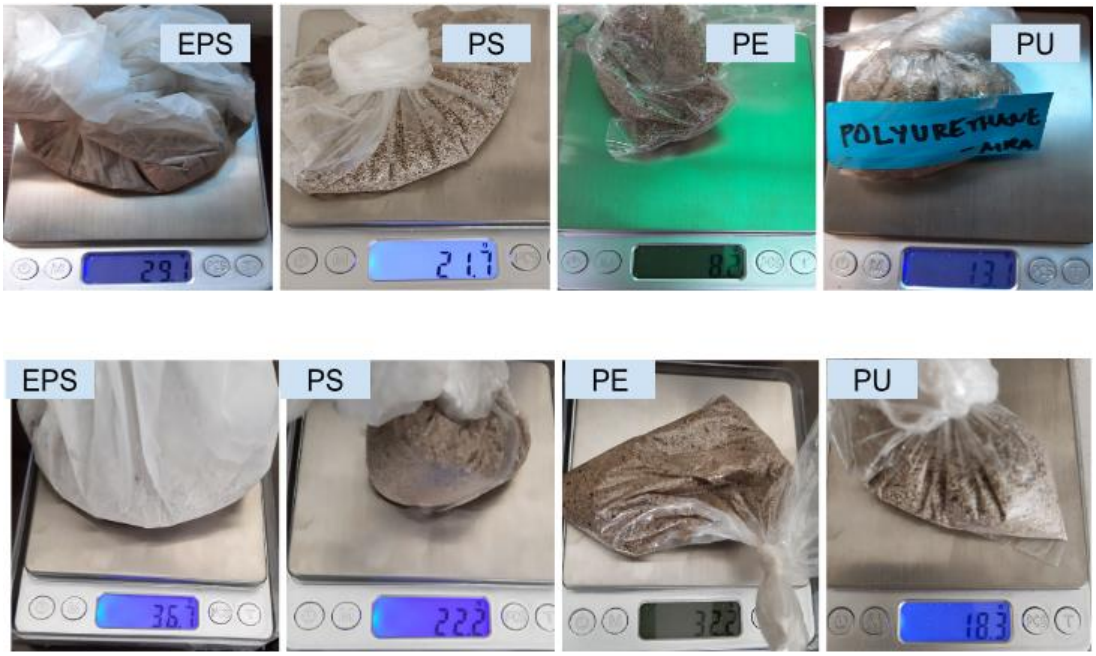


Figure 77. Frass of the superworms in 1 week and in 2 weeks.

Figure 78. Interview with Sir Ledl Lin, an exotic pet owner.

Figure 79. Site visitation with Engr. Madrigal and Ms. De Jose.

Table 33. Interview with Ed Peruelo, a superworm breeder and seller

Questions	**Answers**
How long have you been breeding/selling superworms?	Since 2008, from personal use to business purpose.
Who are your biggest clients?	Mostly pet shops in the earlier years (2008-2010) but when I expanded my business, Manila Ocean Park became a client for a year and became a steady supplier to Cartimar pet shops in Pasay up until now.
At what price do you sell your product?	50 cents to 1 peso from 2008-2010. But today, I sell it at 30 cents.
How many superworms can you produce per month?	Using 50, 000 beetles as breeders, I can produce 500k superwoms per month. Can increase when the climate is cold.
Did you know that Superworms can feed on plastics?	Yes, but I haven't tried it.
Would you sell superworms that feed on plastics?	No, because some pets are expensive and if the superworms that I sold to them caused sickness, my business will be ruined.
Would you invest in a prototype that automates the breeding and segregation of superworms?	Yes because breeding superworms takes up a lot of time and effort since you need to carefully monitor it. Automating it will be a big help for my business.
How long do you think will achieve the ROI if you invest in our prototype?	At a 50k price point, I believe that I can achieve ROI within 2 years as I can reproduce a lot of worms consistently.

To be able to do credible market research, the researchers have also interviewed a consumer, Lendl Lin, an exotic pet owner that houses 100+ species of reptiles and amphibians. He says that the market of superworms is a vast and overlooked business since superworms are very useful and can provide a lot of protein for pets.

Table 34. Interview with Lendl Lin

Questions	**Answers**
How long have you been using superworms as feeds?	I've been using superworms for almost 25 years now, together with crickets and roaches.
What kind of superworms do you usually buy?	I mostly buy jumbo superworms
How many superworms do you usually buy?	About 3000 superworms that ranges to 900 to 1000 pesos
What pets do you have?	I have a lot of pets, mainly reptiles and amphibians that numbers around 100 species
Would you feed your pets superworms that feed on plastics?	If it's proven that it really is safe, why not. I also feed them the carcass of my pets because superworms are clean and will remove the foul odor of dead bodies, and will clean the bones. I do this because I like to store the bones of the animals.
Would you invest in a machine that automates the life of superworms?	Yes, because I'm a firm believer of technology and that will make my life easier as it automates everything.

4.4.1 Comparative Analysis of the Previous Versions of the Project

4.4.1.1 Defining and Implementing various Improvements for the Biodegradation of plastics

The current study used Superworms instead of mealworms due to the capacity of Superworms in their digestive system. And as well as their longevity in biodegrading due to their nature of not undergoing metamorphosis when in a pack. The feeding of the 4 types of plastics in order to further improve the solution of plastic biodegradation.

Hardware and software

The current study changed the hardware and software compared from the previous studies. In the hardware part, an ESP32 is used for the microcontroller compared to the previous studies which used both Arduino Mega and ESP32. This change improved the latency of the transmission of data and remote control of the prototype. The use of a small linear actuator and a DC worm gear to facilitate the transferring of organisms rather than the use of a moving container. For the temperature sensor, the previous studies used DHT22 which was an overkill for the small project, thus in this study, DHT11 was used as it was more than enough for that capability.

These changes improved the mechanisms and were more affordable compared to the previous studies, but did not decrease their quality.

4.4.1.2 Testing Procedure for Data Gathering

Table 35. Testing procedure parameters

Period (weekly)	Plastic	Plastics Consumed (g)
Week 1	PS	39.9
Week 2	PS	38.64
Week 3	EPS	39.23
Week 4	EPS	39.9
Week 5	PE	3.22
Week 6	PE	3.18
Week 7	PU	14.98
Week 8	PU	15.61

Table 36 shows the result of the testing procedure. The data shown above were the weight of plastics before and after Superworms consumed. The total number of Superworms were 4000 and changing of plastic as their sole diet happens every after two weeks or 14 days.

Computation for daily consumption rate per worm:

For Week 1 (EPS)

Daily consumption per worm: $\frac{Total\ Consumption}{7\ days} \times \frac{1}{4000\ worms}$

$= \frac{39.9\ grams}{7\ days} \times \frac{1}{4000\ worms} = 1.425\ mg\ /day\ per\ worm$

For Week 2 (EPS)

Daily consumption per worm: $\frac{Total\ Consumption}{7\ days} \times \frac{1}{4000\ worms}$

$= \frac{38.64\ grams}{7\ days} \times \frac{1}{4000\ worms} = 1.38\ mg\ /day\ per\ worm$

Average: $\frac{W1 + W2}{2} = \frac{1.425 + 1.38}{2} = 1.4025\ mg\ /day\ per\ worm$

For Week 3 (PS)

Daily consumption per worm: $\frac{Total\ Consumption}{7\ days} \times \frac{1}{4000\ worms}$

$= \frac{39.23\ grams}{7\ days} \times \frac{1}{4000\ worms} = 1.401\ mg\ /day\ per\ worm$

For Week 4 (PS)

Daily consumption per worm: $\frac{Total\ Consumption}{7\ days} \times \frac{1}{4000\ worms}$

$= \frac{39.9\ grams}{7\ days} \times \frac{1}{4000\ worms} = 1.425\ mg\ /day\ per\ worm$

Average: $\frac{W1 + W2}{2} = \frac{1.425 + 1.401}{2} = 1.413\ mg\ /day\ per\ worm$

For Week 5 (PE)

Daily consumption per worm: $\frac{Total\ Consumption}{7\ days} \times \frac{1}{4000\ worms}$

$= \frac{3.22\ grams}{7\ days} \times \frac{1}{4000\ worms} = 0.115\ mg\ /day\ per\ worm$

For Week 6 (PE)

Daily consumption per worm: $\frac{Total\ Consumption}{7\ days} \times \frac{1}{4000\ worms}$

$= \frac{3.18\ grams}{7\ days} \times \frac{1}{4000\ worms} = 0.114\ mg\ /day\ per\ worm$

Average: $\frac{W1 + W2}{2} = \frac{1.425 + 1.401}{2} = 0.115\ mg\ /day\ per\ worm$

For Week 7 (PU)

Daily consumption per worm: $\frac{Total\ Consumption}{7\ days} \times \frac{1}{4000\ worms}$

$= \frac{14.98\ grams}{7\ days} \times \frac{1}{4000\ worms} = 2.14\ mg\ /day\ per\ worm$

For Week 8 (PU)

Daily consumption per worm: $\frac{Total\ Consumption}{7\ days} \times \frac{1}{4000\ worms}$

$= \frac{15.61\ grams}{7\ days} \times \frac{1}{4000\ worms} = 2.23\ mg\ /day\ per\ worm$

Average: $\frac{W1 + W2}{2} = \frac{2.14 + 2.23}{2} = 2.185\ mg\ /day\ per\ worm$

4.4.1.3 Data Visualization of the Project

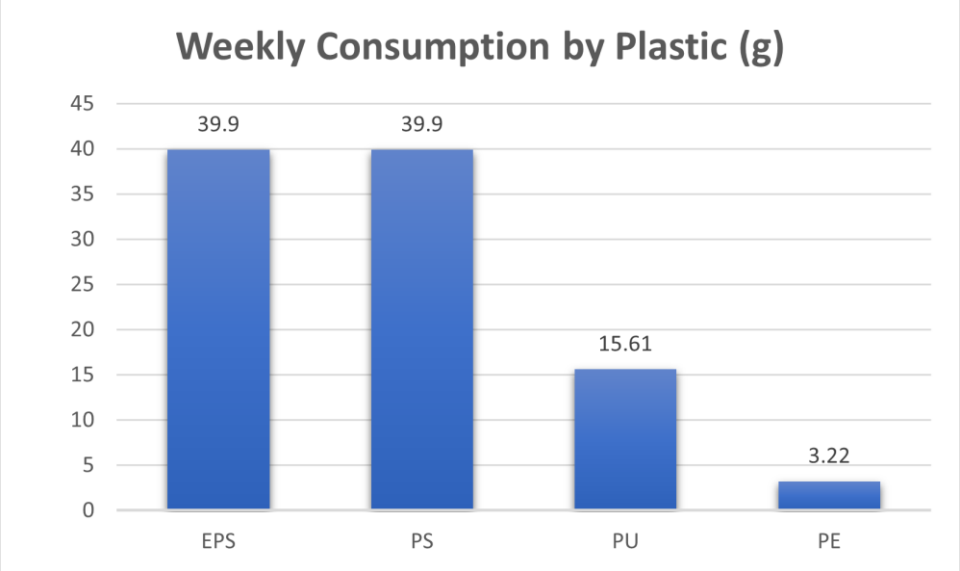


Figure 80. Weekly Consumption of Plastics (g)

Figure 80 shows the weekly consumption of different types of plastics. As shown above, Expanded Polystyrene and Polystyrene were the highest having the same value which is 39.9g, followed by Polyurethane (PU) with a value of 15.61g, and Polyethylene (PE) with 3.22g.

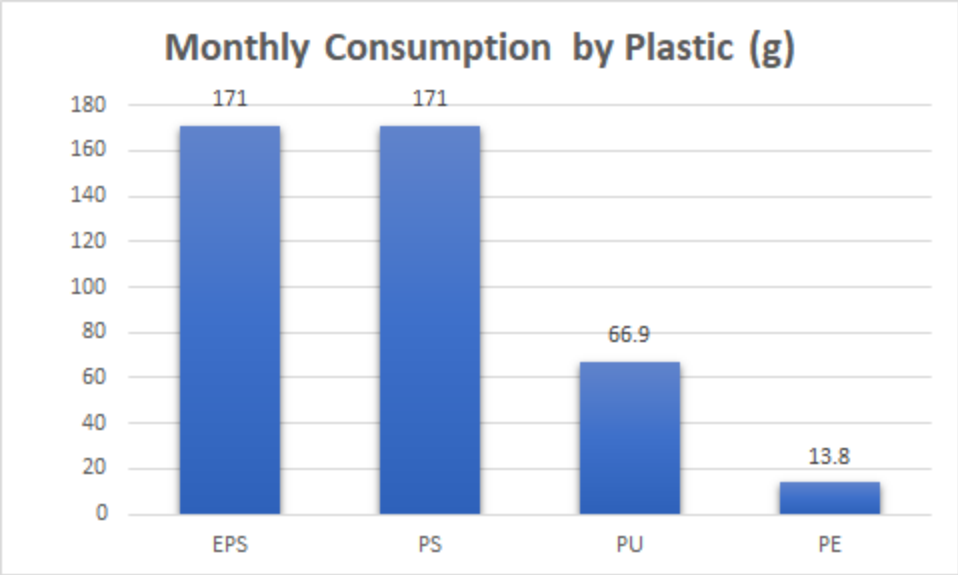


Figure 81. Average Monthly Consumption of Plastics (g)

Figure 81 shows the monthly consumption of Superworms of different types of plastics. Expanded Polystyrene (EPS) and Polystyrene (PS) were the plastics with the highest consumption rate which was 171g, followed by Polyurethane (PU) with a data of 66.9g, Polyethylene is the plastic with the least consumption rate having a data of 13.8g.

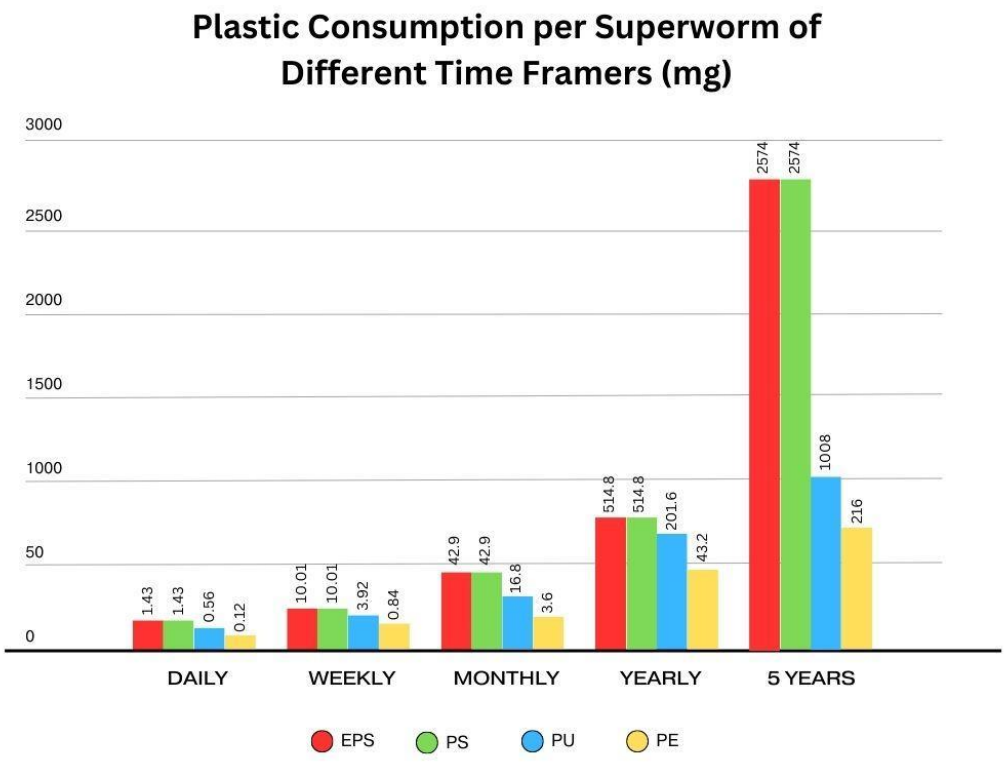


Figure 82. Average Plastic Consumption (mg) in different time frames

Figure 82 shows the data of the consumption rate of Superworms for different types of plastics in different time frames. The types of plastics that were consumed are Expanded Polystyrene (EPS), Polystyrene (PS), Polyurethane (PU), and Polyethylene (PE). The data show how much plastics can Superworms degrade daily, weekly, monthly, yearly, and in 5 years.

Computational Model of Plastic Consumption of Different Time Frames:

Gathered Data (Weekly) (g):

EPS = **10.01g -** The EPS is one of the sole diets of the Superworms and the gathered data within a week was 10.01 grams.

For Monthly Consumption (g):

EPS = Daily Consumption x 30 days

= 1.43 x 30 = **42.9g -** To compute the monthly consumption, we multiplied the computed daily consumption to 30, which indicates the number of days within a month. The calculated monthly consumption was 42.9 grams.

For Yearly Consumption (g):

EPS = Monthly Consumption x 12 months

= 42.9 x 12 = **514.8g -** To compute the yearly consumption, we multiplied the computed monthly consumption to 12, which indicates the number of months within a year. The calculated yearly consumption was 514.8 grams.

In 5 Years Consumption (g):

EPS = Yearly Consumption x 5 years

= 514.8 x 5 years = **2, 574g -** To compute 5 years consumption, we multiplied the computed yearly consumption to 5. The calculated value was 2, 574 grams.

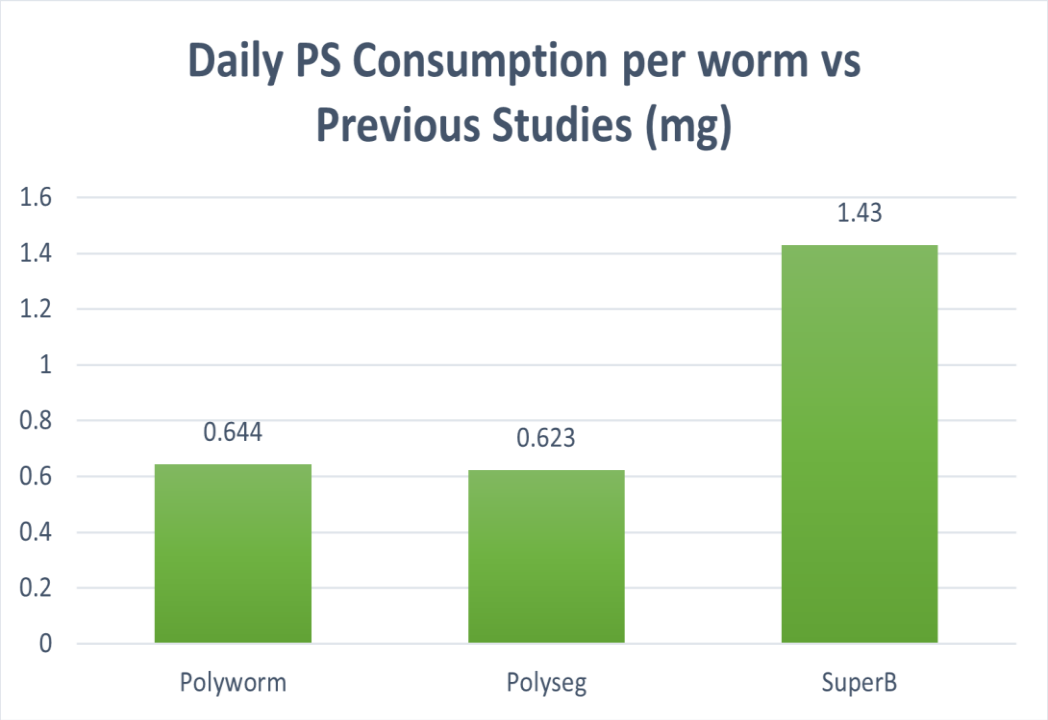


Figure 83. Average Daily PS Consumption per worm vs. Previous Studies (mg)

Figure 83 shows the daily consumption rate of mealworms and superworms regarding polystyrene or styrofoam. As shown in the data, Polyworm consumed 0.644 mg, 0.623 for Polyseg, and 1.43 mg for SuperB. Thus, compared to the previous studies, it was clear that Superworms can degrade Polystyrene more efficiently than Mealworms.

4.4.1.4 Creating a Cross-Tabulation for the Previous Versions of the Project

Table 36. Cross tabulation for previous versions of the project

Project Title	Total Worms Used (pcs)	Daily Consumption per worm (mg)	Consumption of Whole system per month (g)
Automated Superworm Segregation System for Multi-Plastic Biodegradation using ESP32 with Computer Vision for Real-Time monitoring via Mobile Application	4000	1.43	171
Fully Automated Polystyrene Biodegrading System for Mealworm Segregation using Arduino Mega 2560 with ESP 32 for Wireless Notification	1440	0.623	26.9
Semi-Automated Biodegradation System for Polystyrene-Eating Mealworms with ESP8266 for Wireless Notification System	2500	0.644	48.3

With a 1.43 mg of Polystyrene consumption daily, the system consumed 171 g of Polystyrene, as compared to the previous studies which can consume 26.9 up to 48.3 g of Polystyrene in a month.

4.4.2 Establishment of Business to Consumer Model for Marketability of Superworms and its Byproduct

4.4.2.1 Formulating the Profit Percentage

Costs:

Superworms: Php 1200 for 4000 worms

System: 38, 029.00 Php

Total: Php 39, 229.00

Selling Price:

Superworms: Php 300 for 1000 worms

System: Php 50, 000.00

Frass: per 250g - Php 50.00

Selling Price (assuming that we sold 1 system, 8000pcs of superworms, and 500g of frass in a year): Php **52,500.00**

Profit = Sale - Cost

Profit = 52500 - 39229

Profit = Php 13,271.00

Formula:

Profit Percentage = $\frac{Profit}{Cost\ Price}$ x 100

$=\frac{13,271}{39,229}$ x 100

= 33.83%

4.4.2.2 Formulating the expected Return On Investment (ROI)

Table 37. Sample of Simple Cost and Return Table

	Estimated Benefits					Resources needed in Units	Estimated Cost		
No. of Beneficiaries							No. Of Units	Unit Value	Total Cost
Direct Benefits	Year 1	Year 2	Year 3		Direct Cost				
1.Superworm Sales	₱37,500	₱37,500	₱37,500		1.Utilities	kWh/Year	774.84	₱11.32	₱8,771.19
2.Frass Sales	₱1368	₱1368	₱1368						
					Equipment, supplies, and materials				
					1. Sensors and Microcontrollers			₱23,029	₱23,029
					2. Fabrication	pieces	1	₱15,000	₱15,000
					3. Travel	kilometers	Approx 30.8	₱38.96	₱1,200
Total Program Benefits	₱38,868	₱38,868	₱38,868		Opportunity costs				
Benefit-cost ratio	0.78	0.78	0.78		1. Food	Meal/Person	15	₱80	₱1200
Rate of Investment	21.95 %	21.95 %	21.95 %		2. Travel	Person	4	₱150	₱600
Internal Rate of Return (IRR)	58.416%	58.416 %	58.416%						
					Total Program				₱49,800.19

					Costs				

4.4.3 Evaluation of User Acceptance Rating using the ISO standard 25063

4.4.3.1 Evaluate and Validate the Functionality of the Project using Interviews and Evaluation forms

Mark Mercado

Superworm Segregation with Monitoring System using Mobile Application

Introduction: The researchers need to conduct an evaluation questionnaire from the users regarding the different factors and parameters in the functionality of the project entitled as, Automated Superworm Segregation System for Multi-Plastic Biodegradation using ESP32 with Night Vision for Real-Time monitoring via Mobile Application

Instructions: Please rate each of the parameters, and check one response per number based from these ratings; Check 1- if Strongly Disagree, 2- if Disagree,
3 - if Neutral, 4 - if Agree, and 5 - if Strongly Agree.

Survey Statements	Rating				
	1	2	3	4	5
Functional Suitability					
1. The automation of the breeding process of the Superworms is more effective than the traditional method.					✓
2. The automated segregation of the Superworms is more efficient than the manual or traditional method.					✓
3. The live video clearly shows the organisms and ecosystem.				✓	
4. The notification system for the updates on the life stages of the organisms are timely.				✓	
5. The notification system for the harvest of frass in the application is effective.					✓
6. The information for the system's living conditions are well displayed in the interface of the mobile application.					✓
Usability					
7. Both the Hardware and Application are easy to use.			✓		
8. The whole system needs minimum supervision during its entire operation.				✓	
9. The design of the device is pleasing to the eye and accessible.			✓		
Safety and Reliability					
10. The structure of the device is a safe habitat for the organisms.			✓		
11. The construction of the device does not impose hazards to its users.					✓
12. The inlets, outlets, and wiring are intact and properly organized.				✓	

Figure 84. Answered Evaluation form of Mr. Mercado

Jerrney Nena

Superworm Segregation with Monitoring System using Mobile Application					
Introduction: The researchers need to conduct an evaluation questionnaire from the users regarding the different factors and parameters in the functionality of the project entitled as, Automated Superworm Segregation System for Multi-Plastic Biodegradation using ESP32 with Night Vision for Real-Time monitoring via Mobile Application					
Instructions: Please rate each of the parameters, and check one response per number based from these ratings; Check 1- if Strongly Disagree, 2- if Disagree, 3 - if Neutral, 4 - if Agree, and 5 - if Strongly Agree.					
Survey Statements	**Rating**				
	1	**2**	**3**	**4**	**5**
Functional Suitability					
1. The automation of the breeding process of the Superworms is more effective than the traditional method.					✓
2. The automated segregation of the Superworms is more efficient than the manual or traditional method.					✓
3. The live video clearly shows the organisms and ecosystem.				✓	
4. The notification system for the updates on the life stages of the organisms are timely.				✓	
5. The notification system for the harvest of frass in the application is effective.				✓	
6. The information for the system's living conditions are well displayed in the interface of the mobile application.					✓
Usability					
7. Both the Hardware and Application are easy to use.				✓	
8. The whole system needs minimum supervision during its entire operation.				✓	
9. The design of the device is pleasing to the eye and accessible.					✓
Safety and Reliability				✓	
10. The structure of the device is a safe habitat for the organisms.				✓	
11. The construction of the device does not impose hazards to its users.			✓		
12. The inlets, outlets, and wiring are intact and properly organized.					✓

Figure 85. Answered Evaluation form of Mr. Nena

Table 38. Evaluation Rating of the System.

	Evaluators		
	Jernney Nena **Maintenance Officer**	**Mark Mercado** **Maintenance Officer**	
Survey Statements	**Rating**		**Average**
Functional Suitability			
1. The automation of the breeding process of the Superworms is more effective than the traditional method.	5	5	5
2. The automated segregation of the Superworms is more efficient than the manual or traditional method.	5	5	5
3. The live video clearly shows the organisms and ecosystem.	4	4	4
4. The notification system for the updates on the life stages of the organisms are timely.	4	4	4
5. The notification system for the harvest of frass in the application is effective.	4	5	4.5
6. The information for the system's living conditions are well displayed in the interface of the mobile application.	4	5	4.5
Usability			
7. Both the Hardware and Application are easy to use.	4	3	3.5
8. The whole system needs minimum supervision during its entire operation.	4	4	4
9. The design of the device is pleasing to the eye and accessible.	5	3	4
Safety and Reliability			

10. The structure of the device is a safe habitat for the organisms.	4	3	3.5
11. The construction of the device does not impose hazards to its users.	3	5	4
12. The inlets, outlets, and wiring are intact and properly organized.	5	4	4.5

4.5 Project Deployment

The project was initially intended to be deployed in a Material Recovery Facility (MRF) to assist in the biodegradation and decomposition of waste. However, at present, it was deployed at the P3P Homeowners Association for testing purposes. The system was fed with various types of plastics on a weekly basis to measure and record the plastic consumption capacity of the organisms.

During that testing phase, the project aimed to evaluate and assess the efficiency of both the hardware and software components. This included analyzing the performance, reliability, and effectiveness of the different mechanisms, sensors, microcontrollers, and algorithms utilized in the system. The continuous testing process helped identify any issues or areas for improvement, ensuring that the project operated optimally and met its intended goals.

Within the deployment site, the researchers secured an Non-Disclosure Agreement (NDA) and Memorandum of Agreement (MOA) which were received by Jernney Nena

and Mark Mercado, both are community staff in the Homeowners Association of Phase 3 Proper.

Figure 86.

Deployment Site

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

This chapter presents the summary of findings, and reviews the results all throughout the duration of the study. Summarizing all data and providing conclusions as to how the study was developed, and the recommendations of the researchers for future researchers regarding the study of biodegradation of plastics.

5.1 Summary Of Findings

The prototype was developed by the researchers in order to give light on the possible solution of the worsening plastic pollution issue in the world. By optimizing the ability of the Superworms (Zophobas morio) to safely digest and biodegrade plastics, the researchers developed a prototype wherein, it automated the life cycle of the organisms, from breeding up to the segregation of its different life stages namely the Beetle, Pupa, Larvae (worm). This intricate process became a big deal as the Superworms' life cycle is not as straightforward as it with its family, the Mealworms (Tenebrio molitor). Unlike with the Mealworms where the larvae becomes an adult beetle in due time and breeds, a Superworm is needed to be isolated from the rest of the pack of worms to be able to process metamorphosis and become a pupa which will in turn become a beetle. This process takes 2-4 weeks to complete. If not isolated, Superworms will stay in their larvae form for about 6 months to 1 year.

During the study, the researchers fed the superworms 4 different plastics namely, the Polystyrene (PS), Expanded Polystyrene (EPS), Polyethylene (PE), and Polyurethane

(PU). The data gathering for the plastics had a duration of 8 weeks, 2 weeks per plastic. The result showed that the Superworms digest more PS and EPS plastics daily, amounting to 5.7g, followed by the PU with a value of 2.22g and PE with 0.46g.

The study displayed great results for its hardware as it can automate and manually control the mechanisms of the project and maintain the ideal temperature with a range of 27 C to 30 C. It also resulted in a 63% confidence rating for the object detection and counter using the computer vision, to automate the counting of the organisms being produced by the system. However, the study didn't have results in relation with the breeding of the Superworms, even with the 25 beetles produced by the system.

The study also piqued interest among possible investors as the researchers have interviewed a breeder and an exotic pet owner, both saying that they are willing to invest.

The data presented in Chapter 4 showed the efficiency and capability of the prototype in automating the life cycle of the superworms, as well as making it a solution for the worsening plastic pollution.

5.2 Conclusion

The researchers have successfully developed a fully-automated Superworm segregation system for multi-plastic biodegradation using various sensors and mechanical devices, these are the following conclusions that the researchers followed in line to develop the said system:

1. The developed system is using an ESP32 based device that automated the breeding and segregation of Superworms for multi-plastic biodegradation using humidity

and temperature sensor, webcam, weight sensor, servo motor, vibration motor, and linear actuator.

2. The developed mobile application notifies the user, can be used to control parameters, and can monitor the system with computer vision that aids in ensuring the quality of the state of the organisms and the system.
3. The study has determined and compared its efficiency in terms of plastic biodegradation with 122% increase of PS consumption with 0.644 mg from the previous studies, compared to our 1.43 mg PS consumption. Showing its big improvement compared to the previous studies. The system is able to automate the segregation of the life stages using its mechanisms and electronic devices.
4. The researchers have successfully established a business model in maximizing the marketability of the superworms and its byproduct, frass. And has established a firm business idea for possible investors.
5. The system has been tested and evaluated using the ISO standard 25063 by the users of the deployment location.

5.3 Recommendations

1. To further improve the study, researchers recommend that the trays or containers of the superworms are larger and wider to incorporate more organisms, to further improve its objective as a biodegradation solution and business idea.
2. Furthermore, the researchers recommend fully understanding the biology and breeding of the Superworms to be able to continuously produce worms.

3. The researchers also recommend the use of additives and hydrating substances in order to improve the plastic biodegradation.
4. The study can also be improved through the use of a plastic shredder and a better mesh screen flooring, to improve the filtering of scrapped microplastics. All these recommendations will further improve the study and make a better solution for the worsening plastic pollution.

References

[1] Balema, V. P., Hlova, I. Z., Carnahan, S. L., Seyedi, M., Dolotko, O., Rossini, A. J., & Luzinov, I. (2021). Depolymerization of polystyrene under ambient conditions. *New Journal of Chemistry*, *45*(6), 2935–2938. https://doi.org/10.1039/D0NJ05984F

[2] World Bank Group. (2021). Market Study for the Philippines: Plastics Circularity Opportunities and Barriers.

[3] Wehner, M. (2019). It takes decades for polystyrene to break down, even in direct sunlight. Retrieved June 13, 2022, from https://bgr.com/science/how-long-does-plastic-last-polystyrene-waste/

[4] Davis, J. A. (2019). Styrofoam Facts — Why You May Want To Bring Your Own Cup. Retrieved June 13, 2022, from https://www.sej.org/publications/backgrounders/styrofoam-facts-why-you-may-want-bring-your-own-cup#:~:text=of styrofoam in a landfill, put it way beyond that

[5] R. Geyer, J. R. Jambeck, and K. L. Law, "Production, use, and fate of all plastics ever made," Sci. Adv., vol. 3, no. 7, Jul. 2017, doi: 10.1126/sciadv.1700782.

[6] Jiang, S., Su, T., Zhao, J., & Wang, Z. (2021). Biodegradation of Polystyrene by Tenebrio molitor, Galleria mellonella, and Zophobas atratus Larvae and Comparison of Their Degradation Effects. *Polymers*, *13*(20), 3539. https://doi.org/10.3390/polym13203539

[7] Rumbos, C. I., & Athanassiou, C. G. (2021). The Superworm, Zophobas morio (Coleoptera:Tenebrionidae): A 'Sleeping Giant' in Nutrient Sources. *Journal of Insect Science*, *21*(2). https://doi.org/10.1093/jisesa/ieab014

[8] Yang, Y., Wang, J., & Xia, M. (2020). Biodegradation and mineralization of polystyrene by plastic-eating superworms Zophobas atratus. *Science of The Total Environment*, *708*, 135233. https://doi.org/10.1016/j.scitotenv.2019.135233

[9] Brandon, A. M., Gao, S.-H., Tian, R., Ning, D., Yang, S.-S., Zhou, J., … Criddle, C. S. (2018). Biodegradation of Polyethylene and Plastic Mixtures in Mealworms (Larvae of Tenebrio molitor) and Effects on the Gut Microbiome. *Environmental Science & Technology*, *52*(11), 6526–6533. https://doi.org/10.1021/acs.est.8b02301

[10] Lee, H. M., Kim, H. R., Jeon, E., Yu, H. C., Lee, S., Li, J., & Kim, D.-H. (2020). Evaluation of the Biodegradation Efficiency of Four Various Types of Plastics by Pseudomonas aeruginosa Isolated from the Gut Extract of Superworms. *Microorganisms*, *8*(9), 1341. https://doi.org/10.3390/microorganisms8091341

[11] Khan, S., Dong, Y., Nadir, S., Schaefer, D. A., Mortimer, P. E., Xu, J., … Sheng, J. (2021). Valorizing plastic waste by insect consumption. *Circular Agricultural Systems*, *1*(1), 1–9. https://doi.org/10.48130/CAS-2021-0007

[12] G. R. Koerner, Y. G. Hsuan, and R. M. Koerner, "The durability of geosyntPlasticshetics," in Geosynthetics in Civil Engineering, Elsevier, 2007, pp. 36–65. doi: 10.1533/9781845692490.1.36.

[13] B. Petter Jelle, "Nano-based thermal insulation for energy-efficient buildings," in Start-Up Creation, Elsevier, 2016, pp. 129–181. Doi: 10.1016/B978-0-08-100546-0.00008-X.

[14] J. M. Khatib, B. A. Herki, and A. Elkordi, “Characteristics of concrete containing EPS,” in Use of Recycled Plastics in Eco-efficient Concrete, Elsevier, 2019, pp. 137–165. doi: 10.1016/B978-0-08-102676-2.00007-4.

[15] J. H. Wong, M. J. H. Gan, B. L. Chua, M. Gakim, and N. J. Siambun, “Shredder machine for plastic recycling: A review paper,” IOP Conf. Ser. Mater. Sci. Eng., vol. 1217, no. 1, p. 012007, Jan. 2022, doi: 10.1088/1757-899X/1217/1/012007.

[16] Fortune Business Insights, “Polyethylene Market Size, Share & COVID-19 Impact Analysis, By Type (HDPE, LDPE, LLDPE), By End User (Packaging Automotive, Infrastructure & Construction, Consumer Goods/Lifestyle, Healthcare & Pharmaceutical, Electrical & Electronics, Agriculture, Others),” 2020. https://www.fortunebusinessinsights.com/industry-reports/polyethylene-pe-market-101584 (accessed Jun. 15, 2022).

[17] N. T.-H. Pham, “Characterization of Low-Density Polyethylene and LDPE-Based/Ethylene-Vinyl Acetate with Medium Content of Vinyl Acetate,” Polymers (Basel)., vol. 13, no. 14, p. 2352, Jul. 2021, doi: 10.3390/polym13142352.

[18] T. K. Goswami and S. Mangaraj, “Advances in polymeric materials for modified atmosphere packaging (MAP),” in Multifunctional and Nanoreinforced Polymers for Food Packaging, Elsevier, 2011, pp. 163–242. doi: 10.1533/9780857092786.1.163.

[19] J. Bayer, L. A. Granda, J. A. Méndez, M. A. Pèlach, F. Vilaseca, and P. Mutjé, “Cellulose polymer composites (WPC),” in Advanced High Strength Natural Fibre Composites in Construction, Elsevier, 2017, pp. 115–139. doi: 10.1016/B978-0-08-100411-1.00005-4.

[20] S.-C. Li, B. Li, and X.-M. Qin, “Properties of LLDPE Film Modified by LDPE,” Polym. Plast. Technol. Eng., vol. 51, no. 3, pp. 307–310, Feb. 2012, doi: 10.1080/03602559.2011.639833.

[21] H. Li and M. Wang, “Electrospinning and nanofibrous structures for biomedical applications,” in Bioceramics, Elsevier, 2021, pp. 401–436. doi: 10.1016/B978-0-08-102999-2.00018-1.

[22] H. Fałtynowicz, H. Janik, J. Kucinska-Lipka, and M. Sienkiewicz, “Polyurethanes,” in Handbook of Thermoset Plastics, Elsevier, 2022, pp. 231–262. doi: 10.1016/B978-0-12-821632-3.00007-5.

[23] S.-J. Park and M.-K. Seo, “Element and Processing,” 2011, pp. 431–499. doi: 10.1016/B978-0-12-375049-5.00006-2.

[24] A. M. Diez-Pascual, “Preparation and characterization of polyphenylene sulfide nanocomposites,” in Manufacturing of Nanocomposites with Engineering Plastics, Elsevier, 2015, pp. 127–165. doi: 10.1016/B978-1-78242-308-9.00007-0.

[25] Cotton, R. T. (1940). *Mealworms*. U.S. Government Printing Office. Retrieved from https://books.google.com.ph/books?id=JakpGBsoATAC&source=gbs_navlinks_s

[26] Cotton, R. T. (1927). Notes on the Biology of the Meal Worms, Tenebrio Molitor Linne and T. Obscurus Fab. *Annals of the Entomological Society of America, 20*(1), 81–86. https://doi.org/10.1093/aesa/20.1.81

[27] Ribeiro, N., Abelho, M., & Costa, R. (2018). A Review of the Scientific Literature for Optimal Conditions for Mass Rearing Tenebrio molitor (Coleoptera:

Tenebrionidae). *Journal of Entomological Science*, *53*(4), 434–454. https://doi.org/10.18474/JES17-67.1

[28] Carolina. (2010). Superworms Care Sheet. Carolina Biological Supply Company. https://www.carolina.com/teacher-resources/Document/superworms-care-handling-instructions/tr10549.tr

[29] Finke, M. D. (2015). Complete nutrient content of four species of commercially available feeder insects fed enhanced diets during growth. Zoo Biology, 34(6), 554–564. https://doi.org/10.1002/zoo.21246

[30] Tan, Y. (2021). Former BPO employee finds success in farming insects for pet food. https://www.agriculture.com.ph/2021/09/12/former-bpo-employee-finds-success-in-farming-insects-for-pet-food/

[31] Lebel, S. (2021). Kuwaiti breeder eyeing superworms as new superfood. https://manilastandard.net/business/biz-plus/355930/kuwaiti-breeder-eyeing-superworms-as-new-superfood.html

[32] Kuan, Z.-J., Chan, B. K.-N., & Gan, S. K.-E. (2022). Worming the Circular Economy for Biowaste and Plastics: Hermetia illucens, Tenebrio molitor, and Zophobas morio. *Sustainability*, *14*(3), 1594. https://doi.org/10.3390/su14031594

[33] Kwak, K.-W., Kim, S. Y., Ko, H.-J., Lee, K. Y., Song, J.-H., & Yoon, H. J. (2021). Optimal hatching conditions of Zophobas atratus (Coleoptera: Tenebrionidae) eggs under various culture conditions. *Journal of Asia-Pacific Entomology*, *24*(4), 1107–1115. https://doi.org/10.1016/j.aspen.2021.10.006

[34] Maciel-Vergara, G., Jensen, A., & Eilenberg, J. (2018). Cannibalism as a Possible Entry Route for Opportunistic Pathogenic Bacteria to Insect Hosts, Exemplified by Pseudomonas aeruginosa, a Pathogen of the Giant Mealworm Zophobas morio. *Insects*, *9*(3), 88. https://doi.org/10.3390/insects9030088

[35] Peng, B.-Y., Li, Y., Fan, R., Chen, Z., Chen, J., Brandon, A. M. Wu, W.-M. (2020). Biodegradation of low-density polyethylene and polystyrene in superworms, larvae of Zophobas atratus (Coleoptera: Tenebrionidae): Broad and limited extent depolymerization. *Environmental Pollution*, *266*, 115206. https://doi.org/10.1016/j.envpol.2020.115206

[36] Yang, S.-S., Ding, M.-Q., He, L., Zhang, C.-H., Li, Q.-X., Xing, D.-F., … Wu, W.-M. (2021). Biodegradation of polypropylene by yellow mealworms (Tenebrio molitor) and superworms (Zophobas atratus) via gut-microbe-dependent depolymerization. Science of The Total Environment, 756, 144087. https://doi.org/10.1016/j.scitotenv.2020.144087

[37] Luo, L., Wang, Y., Guo, H., Yang, Y., Qi, N., Zhao, X., … Zhou, A. (2021). Biodegradation of foam plastics by Zophobas atratus larvae (Coleoptera: Tenebrionidae) associated with changes of gut digestive enzymes activities and microbiome. *Chemosphere*, *282*, 131006. https://doi.org/10.1016/j.chemosphere.2021.131006

[38] Sun, J., Prabhu, A., Aroney, S., & Rinke, C. (2022). Insights into plastic biodegradation: community composition and functional capabilities of the superworm (Zophobas morio) microbiome in styrofoam feeding trials. BioRxiv. https://doi.org/10.1101/2022.05.16.492041

[39] Arunrattiyakorn, P., Ponprateep, S., Kaennonsang, N., Charapok, Y., Punphuet, Y., Krajangsang, S., … Limtrakul, A. (2022). Biodegradation of

polystyrene by three bacterial strains isolated from the gut of Superworms (Zophobas atratus larvae). Journal of Applied Microbiology, 132(4), 2823–2831. https://doi.org/10.1111/jam.15474

[40] Wang, Y., Luo, L., Li, X., Wang, J., Wang, H., Chen, C., … Zhao, X. (2022). Different Plastics Ingestion Preferences and Efficiencies of Superworm (Zophobas Atratus Fab.) and Mealworm (Tenebrio Molitor Fab.) Associated with Distinct Gut Microbiome Changes. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4028245

[41] Gan, S. K.-E., Phua, S.-X., Yeo, J. Y., Heng, Z. S.-L., & Xing, Z. (2021). Method for Zero-Waste Circular Economy Using Worms for Plastic Agriculture: Augmenting Polystyrene Consumption and Plant Growth. Methods and Protocols, 4(2), 43. https://doi.org/10.3390/mps4020043

[42] Li, J., Kim, H. R., Lee, H. M., Yu, H. C., Jeon, E., Lee, S., & Kim, D.-H. (2020). Rapid biodegradation of polyphenylene sulfide plastic beads by Pseudomonas sp. Science of The Total Environment, 720, 137616. https://doi.org/10.1016/j.scitotenv.2020.137616

[43] Abaya, W. F., Basa, J., Sy, M., Abad, A. C., & Dadios, E. P. (2014). Low cost smart security camera with night vision capability using Raspberry Pi and OpenCV. 2014 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), 1–6. https://doi.org/10.1109/HNICEM.2014.7016253

[44] Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. "YOLOv4: Optimal Speed and Accuracy of Object Detection." *ArXiv:2004.10934 [Cs, Eess]*, April 22, 2020. https://arxiv.org/abs/2004.10934.

[45] Alonzo, R. R., Duñgo, M. G. R., Laspiña, R. R., Lepasana, M. G. D., & Yasuda, A. H. (2020). Semi-Automated Biodegradation System for Polystyrene-Eating Mealworms with ESP8266 for Wireless Notification System. 139.

[46] Migullas, R. M. M., Nabong, R. B. G., Oriol, Q. V. A. D., Santos, M. D. N., & Talde, C. M. M. (2020). Fully Automated Polystyrene Biodegrading System for Mealworm Segregation using Arduino Mega 2560 with ESP 32 for Wireless Notification. 82.

[47] Quennedey, A., Aribi, N., Everaerts, C., & Delbecque, J.-P. (1995). Postembryonic development of Zophobas atratus Fab. (Coleoptera: Tenebrionidae) under crowded or isolated conditions and effects of juvenile hormone analogue applications. *Journal of Insect Physiology*, *41*(2), 143–152. https://doi.org/10.1016/0022-1910(94)00091-T

[48] Tschinkel, W. R. (1993). Crowding, Maternal Age, Age at Pupation, and Life History of Zophobas atratus (Coleoptera: Tenebrionidae). *Annals of the Entomological Society of America*, *86*(3), 278–297. https://doi.org/10.1093/aesa/86.3.278

[49] S. Krishna Chaitanya and K. Dhanalakshmi, "Design and investigation of a linear smart actuator," Apr. 2015, p. 94312T. doi: 10.1117/12.2084502.

[50] O. Makinde, "FUNCTIONALITY ASSESSMENT OF A RECONFIGURABLE VIBRATING SCREEN," TSHWANE Univ. Technol., 2014, doi: 10.13140/RG.2.1.5158.0404.

[51] Pinto, V. H., Gonçalves, J., & Costa, P. (2021). Model of a DC Motor with Worm Gearbox (pp. 638–647). https://doi.org/10.1007/978-3-030-58653-9_61

[52] Baligar, S., Joshi, S., Mudhole, S., Jadhav, S., & Jambotkar, C. (2019). Temperature Based Speed Control of Fan Using Arduino. INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN TECHNOLOGY, 5(10). https://www.researchgate.net/publication/332574596_Temperature_Based_Speed_Control_of_Fan_Using_Arduino

[53] Nigade, A. S., Verma, D., Pandey, B. K., & Srivastav, P. (2016). TEMPERATURE BASED AUTOMATIC FAN SPEED CONTROLLER. International Journal of Current Research, 8(4), 29436–29440. https://www.journalcra.com/sites/default/files/issue-pdf/14313.pdf

[54] M. S. Tiwari, S. M. Hawal, N. N. Mhatre, A. R. Bhosale, and M. Bhaumik, "Automatic Pet Feeder Using Arduino," Int. J. Innov. Res. Sci. Eng. Technol., vol. 7, no. 3, 2018, doi: 10.15680/IJIRSET.2018.0703149.

[55] Masriwilaga, A. A., Al-hadi, T. A. J. M., Subagja, A., & Septiana, S. (2019). Monitoring System for Broiler Chicken Farms Based on Internet of Things (IoT). Telekontran : Jurnal Ilmiah Telekomunikasi, Kendali Dan Elektronika Terapan, 7(1), 1–13. https://doi.org/10.34010/telekontran.v7i1.1641

[56] Ayres, L. B., Lopes, F. S., Garcia, C. D., & Gutz, I. G. R. (2020). Integrated instrumental analysis teaching platform with smartphone-operated fluorometer. Analytical Methods, 12(33), 4109–4115. https://doi.org/10.1039/D0AY01147A

[57] Khawas, C., & Shah, P. (2018). Application of Firebase in Android App Development-A Study. International Journal of Computer Applications, 179(46), 49–53. https://doi.org/10.5120/ijca2018917200

[58] Siegel, D. (2020). There's an App for That, and I Made It. https://journals.sagepub.com/doi/full/10.1177/1076217519880587

[59] Fezari, M., & Dahoud, A. 0Al. (2018). Integrated Development Environment "IDE" For Arduino. https://www.researchgate.net/publication/328615543_Integrated_Development_Environment_IDE_For_Arduino

ANNEX
Appendix A

```
//
**********
// Defines and Includes
//
**********

#include "CMBMenu.hpp"
#include "LiquidCrystal_I2C.h"
#include <Wire.h>
#include <ezButton.h>
#include <DS3231.h>
#include "DHT.h"
#include <FirebaseESP32.h>
#include <WiFi.h>
#include "time.h"
#include <WiFiUdp.h>
#include <NTPClient.h>
#include <HX711_ADC.h>
#include
<Adafruit_PWMServoDriver.h>
#include <EEPROM.h>


#if defined(ESP8266)|| defined(ESP32) ||
defined(AVR)
#include <EEPROM.h>
#endif

#define FIREBASE_HOST "https://test-
9c2e1-default-rtdb.firebaseio.com/"
// #define WIFI_SSID "Converge"
// #define WIFI_PASSWORD
"HDMITWC12438"
// #define WIFI_SSID "Gmad
Fam"
// #define WIFI_PASSWORD
"serafingamad123"
#define FIREBASE_Authorization_key
"mo0YmuGvWEbkgfjBJo4fzUFL6eCzL
zZR4wXTOt0Y"

String SSID ="";
String PASSWORD ="";

//
**********
// Relay Pin Defines
//
**********

#define RELAY_FAN_PIN_4FAN 23 //
#define
RELAY_FAN_PIN_EXHAUSTFAN 14
#define DHTPIN 2
#define DHTTYPE DHT11

// Relay Control Mechanism Pins
uint8_t Relay1_Chamber = 32;
uint8_t Relay2_Chamber = 33;
uint8_t Relay3_4fans = 23;
uint8_t Relay4_ex_fan = 14;
uint8_t Relay5_vib = 25;
uint8_t Relay6_vib = 26;
uint8_t Relay7_dc = 13;
uint8_t Relay8_dc = 12;

// Food Mechanism Pin Assignments
uint8_t stage_1 = 15;
uint8_t stage_2 = 12;
uint8_t stage_3 = 11;
uint8_t stage_4 = 8;

//
**********
// Time Elements
//
**********


const char* ntpServer =
"ph.pool.ntp.org";
const long  gmtOffset_sec = 28800;
```

```
const int daylightOffset_sec = 0;//GMT+8:00

//
**********
// Temperature Thresholds
//
**********

const int TEMP_THRESHOLD_UPPER_4FAN = 35; // NEED NA VALUE THRESHOLD PARA MAG ON
const int TEMP_THRESHOLD_LOWER_4FAN = 32; // lower threshold of temperature, change to your desire value

const int TEMP_THRESHOLD_UPPER_EXHAUSTFAN = 35; // NEED NA VALUE THRESHOLD PARA MAG ON
const int TEMP_THRESHOLD_LOWER_EXHAUSTFAN = 32; // lower threshold of temperature, change to your desire value

//
**********
// Loadcell Pin Defines
//
**********

const int HX711_dout = 16; //mcu > HX711 no 1 dout pin 16
const int HX711_sck = 4; //mcu > HX711 no 1 sck pin 4

const int HX711_dout_2 = 5; //mcu > HX711 no 2 dout pin
const int HX711_sck_2 = 17; //mcu > HX711 no 2 sck pin

const int HX711_dout_3 = 19; //mcu > HX711 no 3 sck pin
const int HX711_sck_3 = 18; //mcu > HX711 no 3 sck pin

//
**********
// Button Pin Defines
//
**********

const int leftButtonPin = 36;
const int rightButtonPin = 39;
const int enterButtonPin = 34;
const int exitButtonPin = 35;
ezButton buttonLeft(leftButtonPin);
ezButton buttonRight(rightButtonPin);
ezButton buttonEnter(enterButtonPin);
ezButton buttonExit(exitButtonPin);

//
**********
// Constructors
//
**********

// Real Time Clock
DS3231 rtc;
// DHT Temperature and Humidity Sensor
DHT dht(DHTPIN, DHTTYPE);
//HX711 constructor (dout pin, sck pin)
HX711_ADC LoadCell(HX711_dout, HX711_sck); //HX711 1
HX711_ADC LoadCell_2(HX711_dout_2, HX711_sck_2); //HX711 2
HX711_ADC LoadCell_3(HX711_dout_3, HX711_sck_3); //HX711 3
// PWM Servomotor Driver
```

```
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();
// Firebase Data Object

FirebaseData firebaseData;
FirebaseData stream;
FirebaseData food;
FirebaseData ssid;
FirebaseData password;

FirebaseJson json;
FirebaseConfig config;
// Menu LIbrary
CMBMenu<100> g_Menu;
// LCD
LiquidCrystal_I2C g_Lcd(0x27, 16, 2);

//
**********
// Variables
//
**********

unsigned long rtc_read, dht_read, rtc_time_check;

int temp1, temp2, humid;

// Load Cell EEPROM Addresses

const int calVal_eepromAdress = 0; // eeprom adress for calibration value load cell 1 (4 bytes)
const int calVal_eepromAdress_2 = 4; // eeprom adress for calibration value load cell 2 (4 bytes)
const int calVal_eepromAdress_3 = 8; // eeprom adress for calibration value load cell 3 (4 bytes)
unsigned long t = 0;

// Servo Motor Min Max Values

const int servoMin = 150; // Minimum pulse width for 0 degree
const int servoMax = 600; // Maximum pulse width for 180 degree

// Control_Mech_variables

int chamber, fan, ex_fan, power, act_rel1, act_rel2;
int vibration1, vibration2, pulley, act_pul1, act_pul2;

// LCD Subentries Variables

int lin_ac_lcd, pull_lcd, ex_fan_lcd, fan_lcd, chamber_lcd, vib1_lcd, vib2_lcd;

int lcd_1st, lcd_2nd, lcd_3rd, lcd_4th ;
int remote, isRemoteOn;

// Food_Mech_Variables

int Stage_1, Stage_2 ,Stage_3, Stage_4;

// Weight Variables
float loadcellVal_2, loadcellVal_3;

// Variables for Exit Buttons per Subentries
static bool Venti_temp_1 = true;
static bool Venti_temp_2 = true;
static bool Venti_cont_fan = true;
static bool Venti_ex_fan = true;

static bool Seg_lin_ac = true;
static bool Seg_pull = true;
static bool Seg_vib1 = true;
static bool Seg_vib2 = true;

static bool Food_1st = true;
static bool Food_2nd = true;
```

```
static bool Food_3rd = true;
static bool Food_4th = true;

static bool time_check = true;
static bool Weight_biocon = true;
static bool Weight_frass = true;

static bool Mobile = true;




// Zero-drift Compensation Variables
float total_weight = 0;
float zero_factor = 0;
float output = 0;
float filtered_output = 0;
const float filter_weight = 0.1;

//
**********
// SUPERB Icon
//
**********


int xx=0;
int yy=0;
byte arrow[8] = {B00000, B00000, B00000, B00000, B11100, B01111, B11100, B00000,};

int T = 200;
byte head1[8] = {
  B10000,
  B11110,
  B11010,
  B11111,
  B11111,
  B11010,
  B11110,
  B10000
};
byte head2[8] = {
  B00001,
  B00011,
  B11111,
  B11111,
  B11111,
  B11111,
  B00011,
  B00001
};
byte anim1[8] = {
  B00000,
  B00000,
  B00001,
  B10011,
  B11111,
  B11111,
  B01110,
  B00100
};
byte anim2[8] = {
  B00100,
  B01110,
  B11111,
  B11111,
  B11011,
  B10001,
  B00000,
  B00000
};
byte tail1[8] = {
  B00000,
  B00000,
  B00011,
  B01111,
  B11111,
  B00011,
  B00000,
  B00000
};
byte tail2[8] = {
  B00000,
  B00000,
  B00011,
  B11111,
  B01111,
  B00011,
  B00000,
  B00000
```

```
};



//
**********
// Text that will be displayed in LCD
//
**********

const char MenuVenti_pc[] PROGMEM = {">Ventilation"};
const char MenuVentiA_pc[] PROGMEM = {">>Temperature"};
const char MenuVentiAa_pc[] PROGMEM = {">>>Board Temp"};
const char MenuVentiAb_pc[] PROGMEM = {">>>BioCon Temp"};
const char MenuVentiB_pc[] PROGMEM = {">>Exhaust Fan"};
const char MenuVentiC_pc[] PROGMEM = {">>Side Fans"};

const char MenuWeight_pc[] PROGMEM = {">Weights"};
const char MenuWeightA_pc[] PROGMEM = {">>BioContainer"};
const char MenuWeightB_pc[] PROGMEM = {">>Frass"};

const char MenuSeg_pc[] PROGMEM = {">Segregation"};
const char MenuSegA_pc[] PROGMEM = {">>Linear Actuator"};
const char MenuSegB_pc[] PROGMEM = {">>Pulley"};
const char MenuSegB_C_pc[] PROGMEM = {">>Vib Motor 1"};
const char MenuSegD_pc[] PROGMEM = {">>Vib Motor 2"};

const char MenuFood_pc[] PROGMEM = {">Food Dispenser"};
const char MenuFoodA_pc[] PROGMEM = {">>Calibrate"};
const char MenuFoodA0_pc[] PROGMEM = {">>>Calibrate LC1"};
const char MenuFoodA1_pc[] PROGMEM = {">>>Calibrate LC2"};
const char MenuFoodB_pc[] PROGMEM = {">>Manual Control"};
const char MenuFoodB0_pc[] PROGMEM = {">>>Load Food"};
const char MenuFoodB1_pc[] PROGMEM = {">>>Send to Cont1"};
const char MenuFoodB2_pc[] PROGMEM = {">>>Send to Cont2"};
const char MenuFoodB3_pc[] PROGMEM = {">>>Send to Cont3"};

const char MenuTime_pc[] PROGMEM = {">Time and Date"};
const char MenuTimeA_pc[] PROGMEM = {">>Display T&D"};
const char MenuTimeB_pc[] PROGMEM = {">>Update Time"};

const char MenuMobile_pc[] PROGMEM = {"> Mobile Mode"};
const char MenuMobileA_pc[] PROGMEM = {">>Open the App"};

//
**********
// Define Function ID's
//
**********
enum MenuFID {
  MenuDummy,
  MenuVenti,
  MenuVentiA,
  MenuVentiAa,
  MenuVentiAb,
  MenuVentiB,
  MenuVentiC,
  MenuWeight,
  MenuWeightA,
```

```
  MenuWeightB,
  MenuSeg,
  MenuSegA,
  MenuSegB,
  MenuSegB_C,
  MenuSegD,
  MenuFood,
  MenuFoodA,
  MenuFoodA0,
  MenuFoodA1,
  MenuFoodB,
  MenuFoodB0,
  MenuFoodB1,
  MenuFoodB2,
  MenuFoodB3,
  MenuTime,
  MenuTimeA,
  MenuTimeB,
  MenuMobile,
  MenuMobileA
};

//
**********
**********
// Define KeyTypes
//
**********
**********

enum KeyType {
  KeyNone, // no key is Pressed
  KeyLeft,
  KeyRight,
  KeyEnter,
  KeyExit
};

//
**********
**********
// Function Prototypes
//
**********
**********
```

```
void VentiAa();
void VentiAb();
void VentiB();
void VentiC();
void WeightA();
void WeightB();
void SegA();
void SegB();
void SegC();
void SegD();
void FoodA();
void FoodA0();
void FoodA1();
void FoodB0();
void FoodB1();
void FoodB2();
void FoodB3();
void TimeA();
void ShutdownA();

void wifi_connect();
void
control_mech_callbackFunc(MultiPathSt
reamData stream);
void
food_mech_callbackFunc(MultiPathStre
amData food);
void menu_loop();
void dht_loop ();
void rtc_loop ();
void food_mech_control ();
void relay_pins ();
void async_functions ();
void loadcell_container_init();
// void loadcell_container_loop();
int bcdToDec(byte bcd);
void segregation_routine ();
void superb_intro();
int bcdToDec(byte bcd);
void checkAndUpdateTimeFromNTP();

void loadcell_2_init();
void loadcell_3_init ();
void loadcell_2_loop();
void loadcell_3_loop();
void loadcell_2and3_loop();
```

```
void loadcell_food_init();
void loadcell_food_loop();

void writeToEEPROM(int address, String str);
String readFromEEPROM(int address);


String parentPath_control = "/CONTROLS_RELAY";
String childPaths_control[8] = {"/POWER", "/FAN", "/EX_FAN", "/PULLEY", "/CHAMBER", "/VIBRATION1", "/VIBRATION2","/MOBILE"};

String parentPath_food = "/FOOD_CONTROL_RELAY";
String childPaths_food[4] = {"/1st Part", "/2nd Part", "/3rd Part", "/4th Part"};


//
**********
// Setup
//
**********
void setup()
{

//
**********
// Initializing Functions
//
**********

  Serial.begin(115200);
  Wire.begin();

  wifi_connect();
  pwm.begin();
  pwm.setPWMFreq(60);
  dht.begin();

//
**********
// Firebase
//
**********

      Firebase.begin(FIREBASE_HOST, FIREBASE_Authorization_key);
  Firebase.reconnectWiFi(true);

  Firebase.beginMultiPathStream(stream, parentPath_control);
  Firebase.setMultiPathStreamCallback(stream, control_mech_callbackFunc, streamTimeoutCallback);
    Firebase.beginMultiPathStream(food, parentPath_food);
  Firebase.setMultiPathStreamCallback(food, food_mech_callbackFunc, foodTimeoutCallback);

  Firebase.getString(ssid, "/SSID");
            Firebase.getString(password, "/Password");
  SSID = ssid.stringData();
  PASSWORD = password.stringData();

  writeToEEPROM(0, SSID);
  writeToEEPROM(0, PASSWORD);


//
**********
// Pinmodes
//
**********

    pinMode(RELAY_FAN_PIN_4FAN, OUTPUT);
```

```
  pinMode(RELAY_FAN_PIN_EXHAUSTFAN, OUTPUT);
                pinMode(Relay1_Chamber, OUTPUT);
  pinMode(Relay2_Chamber, OUTPUT);
  pinMode(Relay3_4fans, OUTPUT);
  pinMode(Relay4_ex_fan, OUTPUT);
  pinMode(Relay5_vib, OUTPUT);
  pinMode(Relay6_vib, OUTPUT);
  pinMode(Relay7_dc, OUTPUT);
  pinMode(Relay8_dc, OUTPUT);


//
**********
// Time Variables for Temperature
//
**********
  dht_read = millis();
  rtc_read = millis();

//
**********
// LCD Menu
//
**********

  g_Lcd.init();
  g_Lcd.backlight();
  g_Lcd.clear();

  superb_intro();

  delay(2000);


   Serial.println("==========");
   Serial.println("SHEEESH SUPER B SHEEESH");
   Serial.println("==========");
  Serial.println("");
  Serial.println("l: left, r: right, e: enter, x: exit, m: print menu");
  Serial.println("");


//
**********
// add nodes to menu (layer, string, function ID)
//
**********


   g_Menu.addNode(0, MenuVenti_pc , MenuVenti);
   g_Menu.addNode(1, MenuVentiA_pc, MenuVentiA);
  g_Menu.addNode(2, MenuVentiAa_pc, MenuVentiAa);
  g_Menu.addNode(2, MenuVentiAb_pc, MenuVentiAb);
   g_Menu.addNode(1, MenuVentiB_pc, MenuVentiB);
   g_Menu.addNode(1, MenuVentiC_pc, MenuVentiC);

  g_Menu.addNode(0, MenuWeight_pc, MenuWeight);
                g_Menu.addNode(1, MenuWeightA_pc, MenuWeightA);
                g_Menu.addNode(1, MenuWeightB_pc, MenuWeightB);

    g_Menu.addNode(0, MenuSeg_pc, MenuSeg);
    g_Menu.addNode(1, MenuSegA_pc, MenuSegA);
    g_Menu.addNode(1, MenuSegB_pc, MenuSegB);
  g_Menu.addNode(1, MenuSegB_C_pc, MenuSegB_C);
    g_Menu.addNode(1, MenuSegD_pc, MenuSegD);
```

```
    g_Menu.addNode(0,    MenuFood_pc,
MenuFood);
   g_Menu.addNode(1,  MenuFoodA_pc,
MenuFoodA);
  g_Menu.addNode(2, MenuFoodA0_pc,
MenuFoodA0);
  g_Menu.addNode(2, MenuFoodA1_pc,
MenuFoodA1);
   g_Menu.addNode(1,  MenuFoodB_pc,
MenuFoodB);
  g_Menu.addNode(2, MenuFoodB0_pc,
MenuFoodB0);
  g_Menu.addNode(2, MenuFoodB1_pc,
MenuFoodB1);
  g_Menu.addNode(2, MenuFoodB2_pc,
MenuFoodB2);
  g_Menu.addNode(2, MenuFoodB3_pc,
MenuFoodB3);

    g_Menu.addNode(0,    MenuTime_pc,
MenuTime);
   g_Menu.addNode(1,  MenuTimeA_pc,
MenuTimeA);
   g_Menu.addNode(1,  MenuTimeB_pc,
MenuTimeB);

   g_Menu.addNode(0,  MenuMobile_pc,
MenuMobile);
                        g_Menu.addNode(1,
MenuMobileA_pc, MenuMobileA);


  // ** menu **
  // build menu and print menu
  // (see terminal for output)
  const char* info;
  g_Menu.buildMenu(info);
  g_Menu.printMenu();

  // ** menu **
  // print current menu entry
  printMenuEntry(info);

  loadcell_food_init();
  loadcell_2_init();
  loadcell_3_init();

}

//
**********
// loop
//
**********
void loop()
{

  if(isRemoteOn) {

  g_Lcd.setCursor(0,0);
  g_Lcd.print("   Mobile Mode  ");
  g_Lcd.setCursor(0,1);
  g_Lcd.print("    Activated   ");

  //DHT CODE FOR LOOPING
    if (millis() - dht_read >= 2000) {
      dht_read = millis();
      dht_loop ();
  }
  //RTC CODE FOR LOOPING
    if (millis() - rtc_read >= 2000) {
      rtc_read = millis();
      rtc_loop ();
  }
  loadcell_2and3_loop();
  loadcell_food_loop();

  } else if (!isRemoteOn){
    menu_loop();
  }

}

//
**********
// ** menu **
// printMenuEntry
```

```
//
**********
void printMenuEntry(const char* f_Info)
{
  String info_s;
        MBHelper::stringFromPgm(f_Info, info_s);

   // when using LCD: add/replace here code to
  // display info on LCD
  Serial.println("----------------");
  Serial.println(info_s);
  Serial.println("----------------");

  // print on LCD
  g_Lcd.clear();
  g_Lcd.setCursor(0, 0);
  g_Lcd.print(info_s);

  // you can print here additional infos into second line of LCD
  // g_Lcd.setCursor(0, 1);
  // g_Lcd.print("my text");
}

void streamTimeoutCallback(bool timeout)
{
  if (timeout)
        Serial.println("stream timed out, resuming...\n");

  if (!stream.httpConnected())
     Serial.printf("error code: %d, reason: %s\n\n", stream.httpCode(), stream.errorReason().c_str());
}

void foodTimeoutCallback(bool timeout)
{
  if (timeout)
         Serial.println("food timed out, resuming...\n");

  if (!food.httpConnected())
     Serial.printf("error code: %d, reason: %s\n\n", food.httpCode(), food.errorReason().c_str());
}

//
**********
// getKey
//
**********
KeyType getKey()
{
  KeyType key = KeyNone;


static bool codeExecuted = false;
  if (!codeExecuted) {
    buttonLeft.setDebounceTime(50);
    buttonRight.setDebounceTime(50);
    buttonEnter.setDebounceTime(50);
    buttonExit.setDebounceTime(50);

    if (buttonLeft.getState()== HIGH) {
      key = KeyLeft;
      Serial.println("Left Button Clicked");
       } else if (buttonRight.getState()== HIGH) {
      key = KeyRight;
             Serial.println("Right Button Clicked");
       } else if (buttonEnter.getState()== HIGH) {
      key = KeyEnter;
             Serial.println("Enter Button Clicked");
       } else if (buttonExit.getState()== HIGH) {
      key = KeyExit;
      Serial.println("Exit Button Clicked");

    }
    codeExecuted = true;
  }
```

```
        if (codeExecuted &&
!(buttonLeft.getState()== HIGH ||
buttonRight.getState()== HIGH ||
buttonEnter.getState()== HIGH ||
buttonExit.getState()== HIGH)) {
   codeExecuted = false;
  }
  return key;
}



void writeToEEPROM(int address,
String str) {
  int length = str.length(); // get the length
of the string
    EEPROM.write(address, length); //
store the length of the string at the given
address

  for (int i = 0; i < length; i++) {
   EEPROM.write(address + i + 1, str[i]);
// store each character of the string in
EEPROM
  }

  EEPROM.commit(); // save changes to
EEPROM
}



String readFromEEPROM(int address) {
  int length = EEPROM.read(address); //
read the length of the string from
EEPROM

  String str = ""; // create an empty string
to hold the retrieved string

  for (int i = 0; i < length; i++) {
    str += char(EEPROM.read(address + i
+ 1)); // retrieve each character of the
string from EEPROM
  }
  return str; // return the retrieved string
}




//
**********
// Ventilations
//
**********


void VentiAa()
{
                Serial.println("Function
MainboardTemp() was called.");

  if (Venti_temp_1){
   while(Venti_temp_1)
   {
    if (millis() - rtc_read >= 1000)
    {
     temp1 = rtc.getTemperature();
     rtc_read = millis();
     g_Lcd.setCursor(0,1);
     g_Lcd.print("  Temp:");
     g_Lcd.setCursor(9,1);
     g_Lcd.print(temp1);
     g_Lcd.setCursor(12,1);
     g_Lcd.print("C");

     buttonExit.loop();
     buttonExit.setDebounceTime(50);
     if (buttonExit.getState()== HIGH) {
      Venti_temp_1 =false;
      Serial.println("----------------");
            Serial.println("Exit   Button
Clicked");
      Serial.println("----------------");
      g_Lcd.setCursor(0,1);
      g_Lcd.print("              ");
     }

    }
```

```
   }
    } else if (!Venti_temp_1 && !(buttonExit.getState()==HIGH)) {
     Venti_temp_1 = true;

  }
}

void VentiAb()
{
                    Serial.println("Function ContainerTemp() was called.");

  if (Venti_temp_2){
   while(Venti_temp_2)
   {
    if (millis() - dht_read >= 1000)
    {
     temp2 = dht.readTemperature();
     humid = dht.readHumidity();

     rtc_read = millis();
     g_Lcd.setCursor(0,1);
     g_Lcd.print("Temp:");
     g_Lcd.setCursor(5,1);
     g_Lcd.print(temp2);
     g_Lcd.setCursor(8,1);
     g_Lcd.print("Humid:");
     g_Lcd.setCursor(14,1);
     g_Lcd.print(humid);


     buttonExit.loop();
     buttonExit.setDebounceTime(50);
     if (buttonExit.getState()== HIGH) {
      Venti_temp_2 =false;
      Serial.println("----------------");
            Serial.println("Exit  Button Clicked");
      Serial.println("----------------");
      g_Lcd.setCursor(0,1);
      g_Lcd.print("              ");
     }
    }
   }
  }
        if (!Venti_temp_2 && !(buttonExit.getState()==HIGH)) {
     Venti_temp_2 = true;
   }
}

void VentiB()
{
                    Serial.println("Function ExhaustFan()  was called.");

 int refresh_rate = 0;

 while(Venti_ex_fan)
 {
  buttonLeft.loop();
  buttonRight.loop();
  buttonExit.loop();
  buttonExit.setDebounceTime(50);
  buttonLeft.setDebounceTime(50);
  buttonRight.setDebounceTime(50);

  //  Exit Function
  if (buttonExit.getState() == HIGH) {
     Venti_ex_fan = false;
     Serial.println("----------------");
            Serial.println("Exit  Button Clicked");
     Serial.println("----------------");
     g_Lcd.setCursor(0,1);
     g_Lcd.print("              ");
     break;
  }

  if(Firebase.ready())
  {
         Firebase.getString(firebaseData, "/CONTROLS_RELAY/EX_FAN");
                    ex_fan_lcd = firebaseData.stringData().toInt();

    if (ex_fan_lcd == 0){
     g_Lcd.setCursor(0,1);
     g_Lcd.print("  ON  >OFF  ");
    } else if (ex_fan_lcd == 1){
     g_Lcd.setCursor(0,1);
```

```
      g_Lcd.print(" >ON    OFF   ");
     }

     // Switch Function
       if (buttonLeft.getState() == HIGH
&& ex_fan_lcd == 0 && millis() -
refresh_rate >= 500) {
           Firebase.setString(firebaseData,
"/CONTROLS_RELAY/EX_FAN",
"1");
       Serial.println("Fan turned ON!");
       refresh_rate = millis();
       } else if (buttonRight.getState() ==
HIGH && ex_fan_lcd == 1 && millis() -
refresh_rate >= 500) {
           Firebase.setString(firebaseData,
"/CONTROLS_RELAY/EX_FAN",
"0");
       Serial.println("Fan turned OFF!");
       refresh_rate = millis();
     }
    }
  }

  // Reset Venti_cont_fan flag to true if
function was exited with Exit button
        if (!Venti_ex_fan &&
!(buttonExit.getState() == HIGH)) {
    Venti_ex_fan = true;
  }
}


void VentiC()
{
  Serial.println("Function ContainerFan()
was called.");

  int refresh_rate = 0;

  while(Venti_cont_fan)
  {
   buttonLeft.loop();
   buttonRight.loop();
   buttonExit.loop();
   buttonExit.setDebounceTime(50);
   buttonLeft.setDebounceTime(50);
   buttonRight.setDebounceTime(50);

   //  Exit Function
   if (buttonExit.getState() == HIGH) {
     Venti_cont_fan = false;
     Serial.println("----------------");
          Serial.println("Exit   Button
Clicked");
     Serial.println("----------------");
     g_Lcd.setCursor(0,1);
     g_Lcd.print("               ");
     break;
   }

   if(Firebase.ready())
   {
         Firebase.getString(firebaseData,
"/CONTROLS_RELAY/FAN");
                        fan_lcd        =
firebaseData.stringData().toInt();

     if (fan_lcd == 0){
      g_Lcd.setCursor(0,1);
      g_Lcd.print("  ON  >OFF   ");
     } else if (fan_lcd == 1){
      g_Lcd.setCursor(0,1);
      g_Lcd.print(" >ON   OFF   ");
     }

     // Switch Function
       if (buttonLeft.getState() == HIGH
&& fan_lcd == 0 && millis() -
refresh_rate >= 500) {
           Firebase.setString(firebaseData,
"/CONTROLS_RELAY/FAN", "1");
       Serial.println("Fan turned ON!");
       refresh_rate = millis();
       } else if (buttonRight.getState() ==
HIGH && fan_lcd == 1 && millis() -
refresh_rate >= 500) {
           Firebase.setString(firebaseData,
"/CONTROLS_RELAY/FAN", "0");
       Serial.println("Fan turned OFF!");
       refresh_rate = millis();
     }
```

```
   }
  }

  // Reset Venti_cont_fan flag to true if function was exited with Exit button
        if (!Venti_cont_fan && !(buttonExit.getState() == HIGH)) {
    Venti_cont_fan = true;
  }
}

//
**********
**********
// Weights
//
**********
**********

void WeightA()
{
                    Serial.println("Function BiodegContainer() was called.");
  static boolean newDataReady_container = 0;
  const int serialPrintInterval_container = 50;

  // zero_factor = LoadCell_2.getData();
  if (Weight_biocon)
  {
   while(Weight_biocon)
   {
    // static float filtered_output_last = 0;
                if (LoadCell_2.update()) newDataReady_container = true;
                if (millis() > t + serialPrintInterval_container) {
                float loadcell_2_data = LoadCell_2.getData();
                  float roundVal_2 = round(loadcell_2_data * 100.0)/ 100.0;
     if (roundVal_2 <0) {
      roundVal_2 = 0;
      }
      g_Lcd.setCursor(0,1);
      g_Lcd.print("Weight: ");
      g_Lcd.setCursor(8,1);
      g_Lcd.print(roundVal_2);
      g_Lcd.setCursor(15,1);
      g_Lcd.print("g");

      buttonExit.loop();
      buttonExit.setDebounceTime(50);
       if (buttonExit.getState()== HIGH)
{
       Weight_biocon =false;
       Serial.println("----------------");
              Serial.println("Exit Button Clicked");
       Serial.println("----------------");
       g_Lcd.setCursor(0,1);
       g_Lcd.print("              ");
       }
      newDataReady_container = 0;
      t = millis();
    }
   }
    } else if (!Weight_biocon && !(buttonExit.getState()==HIGH))
{Weight_biocon = true;}
}


void WeightB()
{
                    Serial.println("Function FrassContainer() was called.");
  static boolean newDataReady_container = 0;
  const int serialPrintInterval_container = 50;

  // zero_factor = LoadCell_2.getData();
  if (Weight_frass)
  {
   while(Weight_frass)
   {
    // static float filtered_output_last = 0;
                if (LoadCell_3.update()) newDataReady_container = true;
```

```
            if (millis() > t + serialPrintInterval_container) {
              float loadcell_3_data = LoadCell_3.getData();
                float roundVal_3 = round(loadcell_3_data * 100.0)/ 100.0;
    if (roundVal_3 <0) {
      roundVal_3 = 0;
      }
      g_Lcd.setCursor(0,1);
      g_Lcd.print("Weight: ");
      g_Lcd.setCursor(8,1);
      g_Lcd.print(roundVal_3);
      g_Lcd.setCursor(15,1);
      g_Lcd.print("g");

      buttonExit.loop();
      buttonExit.setDebounceTime(50);
       if (buttonExit.getState()== HIGH) {
       Weight_frass =false;
       Serial.println("----------------");
            Serial.println("Exit Button Clicked");
       Serial.println("----------------");
       g_Lcd.setCursor(0,1);
       g_Lcd.print("              ");
       }
      newDataReady_container = 0;
      t = millis();
    }
   }
    } else if (!Weight_frass && !(buttonExit.getState()==HIGH)) {Weight_frass = true;}
}


//
**********
// Segregation
//
**********
```

```
void SegA()
{
      Serial.println("Function Linear Actuator() was called.");

 int refresh_rate = 0;

 while(Seg_lin_ac)
 {

  buttonLeft.loop();
  buttonRight.loop();
  buttonExit.loop();
  buttonExit.setDebounceTime(50);
  buttonLeft.setDebounceTime(50);
  buttonRight.setDebounceTime(50);

  //  Exit Function
  if (buttonExit.getState() == HIGH) {
     Seg_lin_ac = false;
     Serial.println("----------------");
           Serial.println("Exit Button Clicked");
     Serial.println("----------------");
     g_Lcd.setCursor(0,1);
     g_Lcd.print("              ");
     break;
  }

  if(Firebase.ready())
  {
        Firebase.getString(firebaseData, "/CONTROLS_RELAY/CHAMBER");
                    lin_ac_lcd = firebaseData.stringData().toInt();

    if (lin_ac_lcd == 0){
      g_Lcd.setCursor(0,1);
      g_Lcd.print("  ON  >OFF  ");
    } else if (lin_ac_lcd == 1){
      g_Lcd.setCursor(0,1);
      g_Lcd.print(" >ON   OFF  ");
    }

    // Switch Function
```

```
      if (buttonLeft.getState() == HIGH && lin_ac_lcd == 0 && millis() - refresh_rate >= 500) {
          Firebase.setString(firebaseData, "/CONTROLS_RELAY/CHAMBER", "1");
          Serial.println("Linear Actuator turned ON!");
      refresh_rate = millis();
      } else if (buttonRight.getState() == HIGH && lin_ac_lcd == 1 && millis() - refresh_rate >= 500) {
          Firebase.setString(firebaseData, "/CONTROLS_RELAY/CHAMBER", "0");
          Serial.println("Linear Actuator turned OFF!");
      refresh_rate = millis();
     }
   }
  }

  // Reset Venti_cont_fan flag to true if function was exited with Exit button
        if (!Seg_lin_ac && !(buttonExit.getState() == HIGH)) {
    Seg_lin_ac = true;
  }
}

void SegB()
{
   Serial.println("Function Pulley was called.");

  int refresh_rate = 0;

  while(Seg_pull)
  {
   buttonLeft.loop();
   buttonRight.loop();
   buttonExit.loop();
   buttonExit.setDebounceTime(50);
   buttonLeft.setDebounceTime(50);
   buttonRight.setDebounceTime(50);

   // Exit Function
   if (buttonExit.getState() == HIGH) {
     Seg_pull = false;
     Serial.println("----------------");
          Serial.println("Exit Button Clicked");
     Serial.println("----------------");
     g_Lcd.setCursor(0,1);
     g_Lcd.print("              ");
     break;
   }

   if(Firebase.ready())
   {
        Firebase.getString(firebaseData, "/CONTROLS_RELAY/PULLEY");
                        pull_lcd = firebaseData.stringData().toInt();

     if (pull_lcd == 0){
      g_Lcd.setCursor(0,1);
      g_Lcd.print("  ON  >OFF  ");
     } else if (pull_lcd == 1){
      g_Lcd.setCursor(0,1);
      g_Lcd.print("  >ON  OFF  ");
     }

     // Switch Function
      if (buttonLeft.getState() == HIGH && pull_lcd == 0 && millis() - refresh_rate >= 500) {
          Firebase.setString(firebaseData, "/CONTROLS_RELAY/PULLEY", "1");
      Serial.println("Fan turned ON!");
      refresh_rate = millis();
      } else if (buttonRight.getState() == HIGH && pull_lcd == 1 && millis() - refresh_rate >= 500) {
          Firebase.setString(firebaseData, "/CONTROLS_RELAY/PULLEY", "0");
      Serial.println("Fan turned OFF!");
      refresh_rate = millis();
     }
   }
```

```
  }

  // Reset Venti_cont_fan flag to true if function was exited with Exit button
  if (!Seg_pull && !(buttonExit.getState() == HIGH)) {
    Seg_pull = true;
  }
}

void SegC()
{
  Serial.println("Function Vibration1 was called.");

  int refresh_rate = 0;

  while(Seg_vib1)
  {
    buttonLeft.loop();
    buttonRight.loop();
    buttonExit.loop();
    buttonExit.setDebounceTime(50);
    buttonLeft.setDebounceTime(50);
    buttonRight.setDebounceTime(50);

    //  Exit Function
    if (buttonExit.getState() == HIGH) {
       Seg_vib1 = false;
       Serial.println("----------------");
                Serial.println("Exit    Button Clicked");
       Serial.println("----------------");
       g_Lcd.setCursor(0,1);
       g_Lcd.print("               ");
       break;
    }

    if(Firebase.ready())
    {
          Firebase.getString(firebaseData, "/CONTROLS_RELAY/VIBRATION1");
                                vib1_lcd = firebaseData.stringData().toInt();
      if (vib1_lcd == 0){
        g_Lcd.setCursor(0,1);
        g_Lcd.print("  ON  >OFF   ");
      } else if (vib1_lcd == 1){
        g_Lcd.setCursor(0,1);
        g_Lcd.print("  >ON   OFF   ");
      }

      // Switch Function
       if (buttonLeft.getState() == HIGH && vib1_lcd == 0 && millis() - refresh_rate >= 500) {
            Firebase.setString(firebaseData, "/CONTROLS_RELAY/VIBRATION1", "1");
          Serial.println("Vibration Motor 1 turned ON!");
        refresh_rate = millis();
       } else if (buttonRight.getState() == HIGH && vib1_lcd == 1 && millis() - refresh_rate >= 500) {
            Firebase.setString(firebaseData, "/CONTROLS_RELAY/VIBRATION1", "0");
          Serial.println("Vibration Motor 1 turned OFF!");
        refresh_rate = millis();
      }
    }
  }

  // Reset Venti_cont_fan flag to true if function was exited with Exit button
          if (!Seg_vib1 && !(buttonExit.getState() == HIGH)) {
    Seg_vib1 = true;
  }
}

void SegD()
{
  Serial.println("Function Vibration2 was called.");

  int refresh_rate = 0;
```

```
  while(Seg_vib2)
  {
   buttonLeft.loop();
   buttonRight.loop();
   buttonExit.loop();
   buttonExit.setDebounceTime(50);
   buttonLeft.setDebounceTime(50);
   buttonRight.setDebounceTime(50);

   //  Exit Function
   if (buttonExit.getState() == HIGH) {
      Seg_vib2 = false;
      Serial.println("----------------");
                Serial.println("Exit    Button
Clicked");
      Serial.println("----------------");
      g_Lcd.setCursor(0,1);
      g_Lcd.print("               ");
      break;
   }

   if(Firebase.ready())
   {
           Firebase.getString(firebaseData,
"/CONTROLS_RELAY/VIBRATION2"
);
                              vib2_lcd        =
firebaseData.stringData().toInt();

     if (vib2_lcd == 0){
      g_Lcd.setCursor(0,1);
      g_Lcd.print("   ON   >OFF   ");
     } else if (vib2_lcd == 1){
      g_Lcd.setCursor(0,1);
      g_Lcd.print(" >ON    OFF   ");
     }

     // Switch Function
       if (buttonLeft.getState() == HIGH
&& vib2_lcd == 0 && millis() -
refresh_rate >= 500) {
           Firebase.setString(firebaseData,
"/CONTROLS_RELAY/VIBRATION2"
, "1");
         Serial.println("Vibration Motor 2
turned ON!");
      refresh_rate = millis();
      } else if (buttonRight.getState() ==
HIGH && vib2_lcd == 1 && millis() -
refresh_rate >= 500) {
          Firebase.setString(firebaseData,
"/CONTROLS_RELAY/VIBRATION2"
, "0");
        Serial.println("Vibration Motor 2
turned OFF!");
      refresh_rate = millis();
     }
    }
  }

  // Reset Venti_cont_fan flag to true if
function was exited with Exit button
          if          (!Seg_vib2          &&
!(buttonExit.getState() == HIGH)) {
   Seg_vib2 = true;
  }
}

//
**********************************
***********
// Food
//
**********************************
***********

void FoodA0()
{

              Firebase.getInt(firebaseData,
"/CALIBRATE/KNOWN_MASS_LC2"
);
       float       known_mass_lc2       =
firebaseData.intData();

  static bool wait1 = true;
  static bool wait2 = true;
     Serial.println("Function      Calibrate
Loadcell2() was called.");
  g_Lcd.setCursor(0,0);
  g_Lcd.print("               ");
  delay(2000);
```

```
  g_Lcd.setCursor(0,0);
  g_Lcd.print("Calibration LC2 ");
  g_Lcd.setCursor(0,1);
  g_Lcd.print("     Start!     ");

  while (wait1) {
    g_Lcd.setCursor(0,0);
    g_Lcd.print("Remove any mass ");
    g_Lcd.setCursor(0,1);
    g_Lcd.print("in the container");
    buttonEnter.loop();
    buttonEnter.setDebounceTime(50);
    if (buttonEnter.getState() == HIGH) {
      wait1 = false;
      g_Lcd.setCursor(0,0);
      g_Lcd.print("                ");
      g_Lcd.setCursor(0,1);
      g_Lcd.print("                ");
    }
  }

  LoadCell_2.tare();
  LoadCell_2.refreshDataSet();

  while (wait2) {
    g_Lcd.setCursor(0,0);
    g_Lcd.print("Put the mass in ");
    g_Lcd.setCursor(0,1);
    g_Lcd.print(" the container. ");

    buttonEnter.loop();
    buttonEnter.setDebounceTime(50);
    if (buttonEnter.getState() == HIGH) {
      wait2 = false;
      g_Lcd.setCursor(0,0);
      g_Lcd.print("                ");
      g_Lcd.setCursor(0,1);
      g_Lcd.print("                ");
    }
  }

  LoadCell_2.refreshDataSet();
       float       newCalibrationValue       =
LoadCell_2.getNewCalibration(known_
mass_lc2);
  delay(1500);
  g_Lcd.setCursor(0,0);
  g_Lcd.print("The new CalVal: ");
  g_Lcd.setCursor(0,1);
  g_Lcd.print(newCalibrationValue);
            Firebase.setString(firebaseData,
"/CALIBRATION_LC2",
newCalibrationValue);

  if (wait1 == false) {
    wait1 = true;

  }
  if (wait2 == false) {
    wait2 = true;
    g_Lcd.setCursor(0,0);
    g_Lcd.print("                ");
    g_Lcd.setCursor(0,1);
    g_Lcd.print("                ");
  }
}

void FoodA1()
{
                Firebase.getInt(firebaseData,
"/CALIBRATE/KNOWN_MASS_LC3"
);
        float       known_mass_lc3       =
firebaseData.intData();

  static bool wait1 = true;
  static bool wait2 = true;
      Serial.println("Function      Calibrate
Loadcell3() was called.");
  g_Lcd.setCursor(0,0);
  g_Lcd.print("                ");
  delay(2000);
  g_Lcd.setCursor(0,0);
  g_Lcd.print("Calibration LC3 ");
  g_Lcd.setCursor(0,1);
  g_Lcd.print("     Start!     ");

  while (wait1) {
    g_Lcd.setCursor(0,0);
    g_Lcd.print("Remove any mass ");
    g_Lcd.setCursor(0,1);
    g_Lcd.print("in the container");
```

```
  buttonEnter.loop();
  buttonEnter.setDebounceTime(50);
  if (buttonEnter.getState() == HIGH) {
    wait1 = false;
    g_Lcd.setCursor(0,0);
    g_Lcd.print("               ");
    g_Lcd.setCursor(0,1);
    g_Lcd.print("               ");
  }
 }

 LoadCell_3.tare();
 LoadCell_3.refreshDataSet();

 while (wait2) {
  g_Lcd.setCursor(0,0);
  g_Lcd.print("Put the mass in ");
  g_Lcd.setCursor(0,1);
  g_Lcd.print(" the container. ");

  buttonEnter.loop();
  buttonEnter.setDebounceTime(50);
  if (buttonEnter.getState() == HIGH) {
    wait2 = false;
    g_Lcd.setCursor(0,0);
    g_Lcd.print("               ");
    g_Lcd.setCursor(0,1);
    g_Lcd.print("               ");
  }
 }

 LoadCell_3.refreshDataSet();
      float newCalibrationValue =
LoadCell_3.getNewCalibration(known_
mass_lc3);
 delay(1500);
 g_Lcd.setCursor(0,0);
 g_Lcd.print("The new CalVal: ");
 g_Lcd.setCursor(0,1);
 g_Lcd.print(newCalibrationValue);
          Firebase.setString(firebaseData,
"/CALIBRATION_LC3",
newCalibrationValue);

 if (wait1 == false) {
  wait1 = true;
 }
 if (wait2 == false) {
  wait2 = true;
  g_Lcd.setCursor(0,0);
  g_Lcd.print("               ");
  g_Lcd.setCursor(0,1);
  g_Lcd.print("               ");
 }

}

void FoodB0()
{
  Serial.println("Function 1stPart() was
called.");

 int refresh_rate = 0;

 while(Food_1st)
 {
  buttonLeft.loop();
  buttonRight.loop();
  buttonExit.loop();
  buttonExit.setDebounceTime(50);
  buttonLeft.setDebounceTime(50);
  buttonRight.setDebounceTime(50);

  //  Exit Function
  if (buttonExit.getState() == HIGH) {
     Food_1st = false;
     Serial.println("----------------");
          Serial.println("Exit Button
Clicked");
     Serial.println("----------------");
     g_Lcd.setCursor(0,1);
     g_Lcd.print("               ");
     break;
  }

  if(Firebase.ready())
  {
        Firebase.getString(firebaseData,
"/FOOD_CONTROL_RELAY/1st
Part");
```

```
                    lcd_1st = firebaseData.stringData().toInt();

    if (lcd_1st == 0){
      g_Lcd.setCursor(0,1);
      g_Lcd.print("  ON  >OFF  ");
    } else if (lcd_1st == 1){
      g_Lcd.setCursor(0,1);
      g_Lcd.print(" >ON   OFF  ");
    }

    // Switch Function
      if (buttonLeft.getState() == HIGH && lcd_1st == 0 && millis() - refresh_rate >= 500) {
          Firebase.setString(firebaseData, "/FOOD_CONTROL_RELAY/1st Part", "1");
      Serial.println("1st Part turned ON!");
      refresh_rate = millis();
      } else if (buttonRight.getState() == HIGH && lcd_1st == 1 && millis() - refresh_rate >= 500) {
          Firebase.setString(firebaseData, "/FOOD_CONTROL_RELAY/1st Part", "0");
          Serial.println("1st Part turned OFF!");
      refresh_rate = millis();
    }
   }
  }

  // Reset Venti_cont_fan flag to true if function was exited with Exit button
          if (!Food_1st && !(buttonExit.getState() == HIGH)) {
    Food_1st= true;
  }
}

void FoodB1()
{
  Serial.println("Function 2ndPart() was called.");
  int refresh_rate = 0;

  while(Food_2nd)
  {
   buttonLeft.loop();
   buttonRight.loop();
   buttonExit.loop();
   buttonExit.setDebounceTime(50);
   buttonLeft.setDebounceTime(50);
   buttonRight.setDebounceTime(50);

   //  Exit Function
   if (buttonExit.getState() == HIGH) {
      Food_2nd = false;
      Serial.println("----------------");
          Serial.println("Exit Button Clicked");
      Serial.println("----------------");
      g_Lcd.setCursor(0,1);
      g_Lcd.print("              ");
      break;
   }

   if(Firebase.ready())
   {
         Firebase.getString(firebaseData, "/FOOD_CONTROL_RELAY/2nd Part");
                    lcd_2nd = firebaseData.stringData().toInt();

     if (lcd_2nd == 0){
      g_Lcd.setCursor(0,1);
      g_Lcd.print("  ON  >OFF  ");
     } else if (lcd_2nd == 1){
      g_Lcd.setCursor(0,1);
      g_Lcd.print(" >ON   OFF  ");
     }

     // Switch Function
      if (buttonLeft.getState() == HIGH && lcd_2nd == 0 && millis() - refresh_rate >= 500) {
          Firebase.setString(firebaseData, "/FOOD_CONTROL_RELAY/2nd Part", "1");
```

```
          Serial.println("2nd Part turned ON!");
       refresh_rate = millis();
       } else if (buttonRight.getState() == HIGH && lcd_2nd == 1 && millis() - refresh_rate >= 500) {
          Firebase.setString(firebaseData, "/FOOD_CONTROL_RELAY/2nd Part", "0");
          Serial.println("2nd Part turned OFF!");
       refresh_rate = millis();
      }
    }
  }

  // Reset Venti_cont_fan flag to true if function was exited with Exit button
          if (!Food_2nd && !(buttonExit.getState() == HIGH)) {
    Food_2nd= true;
  }
}

void FoodB2()
{
  Serial.println("Function 3rdPart() was called.");

  int refresh_rate = 0;

  while(Food_3rd)
  {
   buttonLeft.loop();
   buttonRight.loop();
   buttonExit.loop();
   buttonExit.setDebounceTime(50);
   buttonLeft.setDebounceTime(50);
   buttonRight.setDebounceTime(50);

   //  Exit Function
   if (buttonExit.getState() == HIGH) {
      Food_3rd = false;
      Serial.println("----------------");
          Serial.println("Exit Button Clicked");
      Serial.println("----------------");
      g_Lcd.setCursor(0,1);
      g_Lcd.print("              ");
      break;
   }

   if(Firebase.ready())
   {
          Firebase.getString(firebaseData, "/FOOD_CONTROL_RELAY/3rd Part");
                                lcd_3rd = firebaseData.stringData().toInt();

     if (lcd_3rd == 0){
      g_Lcd.setCursor(0,1);
      g_Lcd.print("  ON  >OFF  ");
     } else if (lcd_3rd == 1){
      g_Lcd.setCursor(0,1);
      g_Lcd.print("  >ON   OFF  ");
     }

     // Switch Function
       if (buttonLeft.getState() == HIGH && lcd_3rd == 0 && millis() - refresh_rate >= 500) {
          Firebase.setString(firebaseData, "/FOOD_CONTROL_RELAY/3rd Part", "1");
          Serial.println("3rd Part turned ON!");
       refresh_rate = millis();
       } else if (buttonRight.getState() == HIGH && lcd_3rd == 1 && millis() - refresh_rate >= 500) {
          Firebase.setString(firebaseData, "/FOOD_CONTROL_RELAY/3rd Part", "0");
          Serial.println("3rd Part turned OFF!");
       refresh_rate = millis();
      }
    }
  }
```

```
  // Reset Venti_cont_fan flag to true if function was exited with Exit button
          if (!Food_3rd && !(buttonExit.getState() == HIGH)) {
    Food_3rd= true;
  }
}




void FoodB3()
{
  Serial.println("Function 4thPart() was called.");

 int refresh_rate = 0;

 while(Food_4th)
 {
  buttonLeft.loop();
  buttonRight.loop();
  buttonExit.loop();
  buttonExit.setDebounceTime(50);
  buttonLeft.setDebounceTime(50);
  buttonRight.setDebounceTime(50);

  //  Exit Function
  if (buttonExit.getState() == HIGH) {
     Food_4th = false;
     Serial.println("----------------");
            Serial.println("Exit Button Clicked");
     Serial.println("----------------");
     g_Lcd.setCursor(0,1);
     g_Lcd.print("              ");
     break;
  }

  if(Firebase.ready())
  {
        Firebase.getString(firebaseData, "/FOOD_CONTROL_RELAY/4th Part");
                          lcd_4th = firebaseData.stringData().toInt();
    if (lcd_4th == 0){
     g_Lcd.setCursor(0,1);
     g_Lcd.print("  ON  >OFF  ");
    } else if (lcd_4th == 1){
     g_Lcd.setCursor(0,1);
     g_Lcd.print("  >ON  OFF  ");
    }

    // Switch Function
     if (buttonLeft.getState() == HIGH && lcd_4th == 0 && millis() - refresh_rate >= 500) {
        Firebase.setString(firebaseData, "/FOOD_CONTROL_RELAY/4th Part", "1");
        Serial.println("4th Part turned ON!");
      refresh_rate = millis();
      } else if (buttonRight.getState() == HIGH && lcd_4th == 1 && millis() - refresh_rate >= 500) {
        Firebase.setString(firebaseData, "/FOOD_CONTROL_RELAY/4th Part", "0");
        Serial.println("4th Part turned OFF!");
      refresh_rate = millis();
     }
   }
 }

  // Reset Venti_cont_fan flag to true if function was exited with Exit button
          if (!Food_4th && !(buttonExit.getState() == HIGH)) {
    Food_4th= true;
  }
}

//
**********
// Time
```

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

void TimeA()
{
  Serial.println("Function  DisplayTime() was called.");

   if (time_check){
    while(time_check)
    {
     if (millis() - rtc_time_check >= 1000)
     {
      byte second, minute, hour, dow, date, month, year;
       bool h12, PM_time, Century;

       second = rtc.getSecond();
       minute = rtc.getMinute();
       hour = rtc.getHour(h12, PM_time);
       dow = rtc.getDoW();
       date = rtc.getDate();
       month = rtc.getMonth(Century);
       year = rtc.getYear();

       g_Lcd.setCursor(0,1);
       g_Lcd.print(hour, DEC);
       g_Lcd.setCursor(2,1);
       g_Lcd.print(":");
       g_Lcd.setCursor(3,1);
       g_Lcd.print(minute, DEC);
       g_Lcd.setCursor(5,1);
       g_Lcd.print(":");
       g_Lcd.setCursor(6,1);
       g_Lcd.print(second, DEC);
       g_Lcd.setCursor(8,1);
       g_Lcd.print(month, DEC);
       g_Lcd.setCursor(10,1);
       g_Lcd.print("/");
       g_Lcd.setCursor(11,1);
       g_Lcd.print(date, DEC);
       g_Lcd.setCursor(13,1);
       g_Lcd.print("/");
       g_Lcd.setCursor(14,1);
       g_Lcd.print(year, DEC);
       buttonExit.loop();
       buttonExit.setDebounceTime(50);
       if (buttonExit.getState()== HIGH) {
         time_check =false;
         Serial.println("----------------");
                Serial.println("Exit   Button Clicked");
         Serial.println("----------------");
         g_Lcd.setCursor(0,1);
         g_Lcd.print("               ");
       }

     }
    }
     } else if (!time_check && !(buttonExit.getState()==HIGH)) {
      time_check = true;
  }
}

void TimeB() {
     Serial.println("Update Time was called.");
  checkAndUpdateTimeFromNTP();
}

//
**********
**********
// Shutdown
//
**********
**********

void MobileA()
{
     Serial.println("Mobile Mode was called.");
  g_Lcd.setCursor(0,1);
  g_Lcd.print("Exit to Stop");
  g_Lcd.setCursor(0,1);
  g_Lcd.print("               ");
```

```
 int refresh_rate = 0;

 while(Mobile)
 {
  buttonLeft.loop();
  buttonRight.loop();
  buttonExit.loop();
  buttonExit.setDebounceTime(50);
  buttonLeft.setDebounceTime(50);
  buttonRight.setDebounceTime(50);

  //  Exit Function
  if (buttonExit.getState() == HIGH) {
     Mobile = false;
     Serial.println("-----------------");
              Serial.println("Exit    Button
Clicked");
     Serial.println("-----------------");
     g_Lcd.setCursor(0,1);
     g_Lcd.print("               ");
     break;
  }

  if(Firebase.ready())
  {
          Firebase.getString(firebaseData,
"/CONTROLS_RELAY/MOBILE");
                          remote          =
firebaseData.stringData().toInt();

    if (remote == 0){
     g_Lcd.setCursor(0,1);
     g_Lcd.print("   ON   >OFF   ");
    } else if (remote == 1){
     g_Lcd.setCursor(0,1);
     g_Lcd.print(" >ON    OFF   ");
    }

    // Switch Function
      if (buttonLeft.getState() == HIGH
&& remote == 0 && millis() -
refresh_rate >= 500) {
          Firebase.setString(firebaseData,
"/CONTROLS_RELAY/MOBILE",
"1");
      Serial.println("Mobile Mode Turned
On!");
      refresh_rate = millis();
      } else if (buttonRight.getState() ==
HIGH && remote == 1 && millis() -
refresh_rate >= 500) {
          Firebase.setString(firebaseData,
"/CONTROLS_RELAY/MOBILE",
"0");
      Serial.println("Mobile Mode Turned
Off!");
      refresh_rate = millis();
     }
   }
  }

  // Reset Venti_cont_fan flag to true if
function was exited with Exit button
  if (!Mobile && !(buttonExit.getState()
== HIGH)) {
    Mobile= true;
  }


}


//
============================
============================
=============
// naming convention
// prefix:
//   g  : global variable
//   f  : function/method variable
//   m  : private member of class
//      : without prefix: public member of
class
//
//
//
============================
============================
=============
```

```
void menu_loop (){

  buttonLeft.loop();
  buttonRight.loop();
  buttonEnter.loop();
  buttonExit.loop();

  // function ID
  int fid = 0;

  // info text from menu
  const char* info;

  // go to deeper or upper layer?
  bool layerChanged=false;

  //



  // determine Pressed key
  KeyType key = getKey();

  // ** menu **
  // call menu methods regarding Pressed
key
  switch(key) {
    case KeyExit:
      g_Menu.exit();
      break;
    case KeyEnter:
      g_Menu.enter(layerChanged);
      break;
    case KeyRight:
      g_Menu.right();
      break;
    case KeyLeft:
      g_Menu.left();
      break;
    default:
      break;

  }

  // ** menu **
    // pint/update menu when key was
Pressed
  // and get current function ID "fid"
  if (KeyNone != key) {
    fid = g_Menu.getInfo(info);
    printMenuEntry(info);
  }


  // ** menu **
  // do action regarding function ID "fid"
  if ((0 != fid) && (KeyEnter == key) &&
(!layerChanged)) {
    switch (fid) {
      case MenuVentiAa:
        VentiAa();
        break;
      case MenuVentiAb:
        VentiAb();
        break;
      case MenuVentiB:
        VentiB();
        break;
      case MenuVentiC:
        VentiC();
        break;
      case MenuWeightA:
        WeightA();
        break;
      case MenuWeightB:
        WeightB();
        break;
      case MenuSegA:
        SegA();
        break;
      case MenuSegB:
        SegB();
        break;
      case MenuSegB_C:
        SegC();
        break;
      case MenuSegD:
        SegD();
        break;
      case MenuFoodA0:
        FoodA0();
        break;
      case MenuFoodA1:
```

```
      FoodA1();
      break;
    case MenuFoodB0:
      FoodB0();
      break;
    case MenuFoodB1:
      FoodB1();
      break;
    case MenuFoodB2:
      FoodB2();
      break;
    case MenuFoodB3:
      FoodB3();
      break;
    case MenuTimeA:
      TimeA();
      break;
    case MenuTimeB:
      TimeB();
      break;
    case MenuMobileA:
      MobileA();
      break;
    default:
      break;
   }
  }

}



void menu_loop (){

  buttonLeft.loop();
  buttonRight.loop();
  buttonEnter.loop();
  buttonExit.loop();

  // function ID
  int fid = 0;

  // info text from menu
  const char* info;

  // go to deeper or upper layer?
  bool layerChanged=false;

  //


  // determine Pressed key
  KeyType key = getKey();

  // ** menu **
  // call menu methods regarding Pressed
key
  switch(key) {
    case KeyExit:
      g_Menu.exit();
      break;
    case KeyEnter:
      g_Menu.enter(layerChanged);
      break;
    case KeyRight:
      g_Menu.right();
      break;
    case KeyLeft:
      g_Menu.left();
      break;
    default:
      break;

  }

  // ** menu **
    // pint/update menu when key was
Pressed
  // and get current function ID "fid"
  if (KeyNone != key) {
    fid = g_Menu.getInfo(info);
    printMenuEntry(info);
  }


  // ** menu **
  // do action regarding function ID "fid"
  if ((0 != fid) && (KeyEnter == key) &&
(!layerChanged)) {
    switch (fid) {
      case MenuVentiAa:
        VentiAa();
```

```
  break;
case MenuVentiAb:
  VentiAb();
  break;
case MenuVentiB:
  VentiB();
  break;
case MenuVentiC:
  VentiC();
  break;
case MenuWeightA:
  WeightA();
  break;
case MenuWeightB:
  WeightB();
  break;
case MenuSegA:
  SegA();
  break;
case MenuSegB:
  SegB();
  break;
case MenuSegB_C:
  SegC();
  break;
case MenuSegD:
  SegD();
  break;
case MenuFoodA0:
  FoodA0();
  break;
case MenuFoodA1:
  FoodA1();
  break;
case MenuFoodB0:
  FoodB0();
  break;
case MenuFoodB1:
  FoodB1();
  break;
case MenuFoodB2:
  FoodB2();
  break;
case MenuFoodB3:
  FoodB3();
  break;
      case MenuTimeA:
        TimeA();
        break;
      case MenuTimeB:
        TimeB();
        break;
      case MenuMobileA:
        MobileA();
        break;
      default:
        break;
    }
  }

}


void menu_loop (){

  buttonLeft.loop();
  buttonRight.loop();
  buttonEnter.loop();
  buttonExit.loop();

  // function ID
  int fid = 0;

  // info text from menu
  const char* info;

  // go to deeper or upper layer?
  bool layerChanged=false;

  //


  // determine Pressed key
  KeyType key = getKey();

  // ** menu **
  // call menu methods regarding Pressed
key
  switch(key) {
    case KeyExit:
      g_Menu.exit();
      break;
```

```
    case KeyEnter:
      g_Menu.enter(layerChanged);
      break;
    case KeyRight:
      g_Menu.right();
      break;
    case KeyLeft:
      g_Menu.left();
      break;
    default:
      break;

  }

  // ** menu **
   // pint/update menu when key was
Pressed
  // and get current function ID "fid"
  if (KeyNone != key) {
    fid = g_Menu.getInfo(info);
    printMenuEntry(info);
  }



  // ** menu **
  // do action regarding function ID "fid"
  if ((0 != fid) && (KeyEnter == key) &&
(!layerChanged)) {
    switch (fid) {
      case MenuVentiAa:
        VentiAa();
        break;
      case MenuVentiAb:
        VentiAb();
        break;
      case MenuVentiB:
        VentiB();
        break;
      case MenuVentiC:
        VentiC();
        break;
      case MenuWeightA:
        WeightA();
        break;
      case MenuWeightB:
        WeightB();
        break;
      case MenuSegA:
        SegA();
        break;
      case MenuSegB:
        SegB();
        break;
      case MenuSegB_C:
        SegC();
        break;
      case MenuSegD:
        SegD();
        break;
      case MenuFoodA0:
        FoodA0();
        break;
      case MenuFoodA1:
        FoodA1();
        break;
      case MenuFoodB0:
        FoodB0();
        break;
      case MenuFoodB1:
        FoodB1();
        break;
      case MenuFoodB2:
        FoodB2();
        break;
      case MenuFoodB3:
        FoodB3();
        break;
      case MenuTimeA:
        TimeA();
        break;
      case MenuTimeB:
        TimeB();
        break;
      case MenuMobileA:
        MobileA();
        break;
      default:
        break;
    }
  }

}
```

```
void dht_loop () {

  humid = dht.readHumidity();
  temp1 = dht.readTemperature();


// Prevent Negative Numbers from displaying
  if (humid <0) {
    humid = 0;
    }
  if (temp1 <0) {
    temp1 = 0;
    }


                Firebase.setInt(firebaseData, "/TEMPERATURE_CONTAINER", temp1);

                Firebase.setInt(firebaseData, "/HUMIDITY", humid);

                  if(temp1 >= TEMP_THRESHOLD_UPPER_4FAN ){

    digitalWrite(RELAY_FAN_PIN_4FAN, LOW); // turn on
        } else if(temp1 <= TEMP_THRESHOLD_LOWER_4FAN) {
    digitalWrite(RELAY_FAN_PIN_4FAN, HIGH); // turn on
  }
}

void dht_loop () {

  humid = dht.readHumidity();
  temp1 = dht.readTemperature();


// Prevent Negative Numbers from displaying
  if (humid <0) {
    humid = 0;
    }
  if (temp1 <0) {
    temp1 = 0;
    }


                Firebase.setInt(firebaseData, "/TEMPERATURE_CONTAINER", temp1);

                Firebase.setInt(firebaseData, "/HUMIDITY", humid);

                  if(temp1 >= TEMP_THRESHOLD_UPPER_4FAN ){

    digitalWrite(RELAY_FAN_PIN_4FAN, LOW); // turn on
        } else if(temp1 <= TEMP_THRESHOLD_LOWER_4FAN) {
    digitalWrite(RELAY_FAN_PIN_4FAN, HIGH); // turn on
  }
}



void rtc_loop () {

  temp2 = rtc.getTemperature();

  if (temp2 <0) {temp2 = 0;}
                Firebase.setInt(firebaseData, "/TEMPERATURE_MAINBOARD", temp2);

                  if(temp2 >= TEMP_THRESHOLD_UPPER_EXHAUSTFAN ){

     digitalWrite(RELAY_FAN_PIN_EXHAUSTFAN, LOW); // turn on
```

```
        } else if(temp2 <= TEMP_THRESHOLD_LOWER_EXHAUSTFAN){
     digitalWrite(RELAY_FAN_PIN_EXHAUSTFAN, HIGH); // turn on
  }

}

void wifi_connect(){

  Serial.print("Connecting to ");
  Serial.println(SSID);
                  WiFi.begin(SSID.c_str(), PASSWORD.c_str());
         while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(300);
  }
  Serial.println("");
  Serial.print("IP Address: ");
  Serial.println(WiFi.localIP());
  Serial.println();
  Serial.print("Starting System");
  Serial.println("");

}



void superb_intro() {
   // initialize g_Lcd and set up the number of columns and rows:

  Wire.begin();
  g_Lcd.init();
  g_Lcd.backlight();
  g_Lcd.createChar(1, arrow);

  for (yy=0; yy<2; yy++) {
    for (xx=0; xx<16; xx++) {
      g_Lcd.setCursor(xx, yy);
      g_Lcd.print(char(1));
      delay(170);
      g_Lcd.setCursor(xx, yy);
      if (yy==0) {
       g_Lcd.print("   SUPERB   "[xx]);
      } else {
                        g_Lcd.print("Breeding Machine"[xx]);
      }
    }
  }
  delay(1000);
   g_Lcd.clear();

  delay(1000);

  g_Lcd.createChar(1, head1);
  g_Lcd.createChar(2, head2);
  g_Lcd.createChar(3, anim1);
  g_Lcd.createChar(4,  anim2);
  g_Lcd.createChar(5, tail1);
  g_Lcd.createChar(6, tail2);
  g_Lcd.setCursor(0,0);
  g_Lcd.write(1);
  delay(T);
  g_Lcd.setCursor(0,0);
  g_Lcd.write(2);
  g_Lcd.write(1);
  delay(T);
  g_Lcd.setCursor(0,0);
  g_Lcd.write(3);
  g_Lcd.write(2);
  g_Lcd.write(1);
  delay(T);
  for (int a=0; a<18; a++)
  {
  g_Lcd.setCursor(a,0);
  g_Lcd.write(5);
  g_Lcd.write(3);
  g_Lcd.write(2);
  g_Lcd.write(1);
  delay(T);
  g_Lcd.clear();
  g_Lcd.setCursor(a+1,0);
  g_Lcd.write(6);
  g_Lcd.write(4);
  g_Lcd.write(2);
  g_Lcd.write(1);
  delay(T);
```

```
  g_Lcd.clear();
  a++;
  }

}

void loadcell_2_loop() {

  static boolean newDataReady_container = 0;
  const int serialPrintInterval_container = 1000; //increase value to slow down serial print activity

                if (LoadCell_2.update()) newDataReady_container = true;
                if (millis() > t + serialPrintInterval_container) {

                float loadcell_2_data = LoadCell_2.getData();
                  float roundVal_2 = round(loadcell_2_data * 100.0)/ 100.0;
     if (roundVal_2 <0) {
       roundVal_2 = 0;
       }
       g_Lcd.setCursor(0,1);
       g_Lcd.print("Weight: ");
       g_Lcd.setCursor(8,1);
       g_Lcd.print(roundVal_2);
       g_Lcd.setCursor(15,1);
       g_Lcd.print("g");
     }
}

void loadcell_2_loop() {

  static boolean newDataReady_container = 0;
  const int serialPrintInterval_container = 1000; //increase value to slow down serial print activity

                if (LoadCell_2.update()) newDataReady_container = true;
                if (millis() > t + serialPrintInterval_container) {

                float loadcell_2_data = LoadCell_2.getData();
                  float roundVal_2 = round(loadcell_2_data * 100.0)/ 100.0;
     if (roundVal_2 <0) {
       roundVal_2 = 0;
       }
       g_Lcd.setCursor(0,1);
       g_Lcd.print("Weight: ");
       g_Lcd.setCursor(8,1);
       g_Lcd.print(roundVal_2);
       g_Lcd.setCursor(15,1);
       g_Lcd.print("g");
     }
}

void checkAndUpdateTimeFromNTP() {

  WiFiUDP ntpUDP;
                                                //
rtc.setEpoch(timeClient.getEpochTime()
);
       NTPClient timeClient(ntpUDP, ntpServer, gmtOffset_sec, daylightOffset_sec);
  timeClient.begin();
  timeClient.update();
  DateTime rtcTime = RTClib::now();

          if (rtcTime.day() == timeClient.getDay() &&
                    rtcTime.hour() == timeClient.getHours() &&
                  rtcTime.minute() == timeClient.getDay()) {
    Serial.println("Time is up to date");
    g_Lcd.setCursor(0,1);
    g_Lcd.print("Time Up to Date");
    delay(1000);
    g_Lcd.print("               ");
  } else {
       Serial.println("Time needs to be updated");
```

```
    Serial.println("Updating...");

    g_Lcd.setCursor(0,1);
    g_Lcd.print("Updating Time...");
    delay(3000);
    g_Lcd.setCursor(0,1);
    g_Lcd.print("                ");

    timeClient.update();
     rtc.setEpoch(timeClient.getEpochTim
e());
    g_Lcd.setCursor(0,1);
    g_Lcd.print("Time Updated");
    delay(2000);
    g_Lcd.setCursor(0,1);
    g_Lcd.print("                ");
    timeClient.end();
  }
}

int bcdToDec(byte bcd) {
  return (bcd / 16 * 10) + (bcd % 16);
}



void loadcell_3_init (){
  LoadCell_3.begin();
   float calibrationValue_3; // calibration
value load cell 2
     calibrationValue_3 = 217.16; //
uncomment this if you want to set this
value in the sketch
  #if defined(ESP8266)|| defined(ESP32)
  #endif
  unsigned long stabilizingtime = 2000; //
preciscion right after power-up can be
improved by adding a few seconds of
stabilizing time
  boolean _tare = true;
  LoadCell_3.start(stabilizingtime, _tare);
  byte loadcell_3_rdy = 0;

     while ((loadcell_3_rdy) < 1) { //run
startup, stabilization and tare, both
modules simultaniously
      if (!loadcell_3_rdy) loadcell_3_rdy =
LoadCell_3.startMultiple(stabilizingtime
, _tare);
    }
    if (LoadCell_3.getTareTimeoutFlag())
{
          Serial.println("Timeout, check
MCU>HX711 no.3 wiring and pin
designations");
    }
    LoadCell_3.setCalFactor(calibrationV
alue_3); // user set calibration value
(float)

}


void loadcell_food_loop() {

 // static boolean newDataReady_food =
0;
const int serialPrintInterval_food = 0;
//increase value to slow down serial print
activity
static boolean pauseReading = false;
const float thresholdWeight = 25.0;
// static boolean pauseRepolyo = false;
float roundVal_i;
float stableVal = 0;
float range = 0.5; // set the range for stable
readings
float i;

unsigned long lastStableCheckTime = 0;
const unsigned long stableInterval =
2000; // 2 seconds in seconds

if (pauseReading) {
  Serial.print("Food Open");
  Serial.println(" ");
   pwm.setPWM(stage_2, 0, map(60, 0,
180, servoMin, servoMax));
  delay(5000);
  Serial.print("Food Close");
  Serial.println(" ");
```

```
  pwm.setPWM(stage_2, 0, map(180, 0, 180, servoMin, servoMax));
  pauseReading = false;
  roundVal_i = 0;
  i = 0;
  delay(5000);
} else {
  if (LoadCell.update() && millis() > t + serialPrintInterval_food) {
    i = LoadCell.getData();
    roundVal_i = round(i * 100.0)/ 100.0;
    if (roundVal_i < 0) {
      roundVal_i = 0;
    }

            Firebase.setFloat(firebaseData, "/WEIGHTS/FOOD", roundVal_i);
    // Serial.print("Load_cell output: ");
    // Serial.println(roundVal_i);

    t = millis();

    if (roundVal_i >= thresholdWeight) {
    pauseReading = true;
    roundVal_i = 0;
    i = 0;
    }
  }
}



  // receive command from serial terminal, send 't' to initiate tare operation:
  if (Serial.available() > 0) {
    char inByte = Serial.read();
              if (inByte == 't') LoadCell.tareNoDelay();
  }

    // check if last tare operation is complete:
  if (LoadCell.getTareStatus() == true) {
    Serial.println("Tare complete for Food Load Cell");
  }
}

// bool isStable(float val1, float val2, float range) {
//   return abs(val1 - val2) < range;
// }


void loadcell_food_init() {

  LoadCell.begin();

  float calibrationValue;
  calibrationValue = 424.67;
  #if defined(ESP8266)|| defined(ESP32)

  #endif

  unsigned long stabilizingtime = 2000; // precisicion right after power-up can be improved by adding a few seconds of stabilizing time
  boolean _tare = true;
  LoadCell.start(stabilizingtime, _tare);
  if (LoadCell.getTareTimeoutFlag()) {
          Serial.println("Timeout, check MCU>HX711 wiring and pin designations");
    while (1);
  }
  else {
     LoadCell.setCalFactor(calibrationValue); // set calibration value (float)
    Serial.println("Startup is complete");
  }
}


void loadcell_2and3_loop(){


  static boolean newDataReady_container = 0;
```

```
  const int serialPrintInterval_container = 1000; //increase value to slow down serial print activity

   // check for new data/start next conversion:

  // LoadCell_2.update();
  // LoadCell_3.update();


        if (LoadCell_2.update() || LoadCell_3.update()) {newDataReady_container = true;}
  //get smoothed value from data set
  if ((newDataReady_container)) {
            if (millis() > t + serialPrintInterval_container) {
     float b = LoadCell_2.getData();
     float c = LoadCell_3.getData();
     // Convert to Hundredth Decimal
     float roundVal_b = round(b * 100.0)/ 100.0;
     float roundVal_c = round(c * 100.0)/ 100.0;
      // Prevent Negative Numbers from displaying
     if (roundVal_b <0) {
       roundVal_b = 0;
     }
     if (roundVal_c <0) {
       roundVal_c = 0;
     }
     // Serial.print("    Load_cell 2 output val: ");
     // Serial.print(b);
     // Serial.print("    Load_cell 3 output val: ");
     // Serial.println(c);
     // Firebase Code
            Firebase.setFloat(firebaseData, "/WEIGHTS/BIOCON", roundVal_b);
            Firebase.setFloat(firebaseData, "/WEIGHTS/FRASS", roundVal_c);
     newDataReady_container = 0;
     t = millis();
    }
  }

}


void loadcell_3_loop() {

  static boolean newDataReady_container = 0;
  const int serialPrintInterval_container = 1000; //increase value to slow down serial print activity

                if (LoadCell_3.update()) newDataReady_container = true;
                if (millis() > t + serialPrintInterval_container) {

       // output = LoadCell_2.getData() - zero_factor;
       // filtered_output = filter_weight * output + (1 - filter_weight) * filtered_output_last;
                // filtered_output_last = filtered_output;

     //Zero-drift compensation:

            // if (abs(filtered_output - filtered_output_last) > 0.03) { //Detect significant changes in the output
         // zero_factor = zero_factor + (filtered_output_last - filtered_output);
            // filtered_output_last = filtered_output;
      // }
             float loadcell_3_data = LoadCell_3.getData();
                float roundVal_3 = round(loadcell_3_data * 100.0)/ 100.0;
     if (roundVal_3 <0) {
       roundVal_3 = 0;
       }
       g_Lcd.setCursor(0,1);
       g_Lcd.print("Weight: ");
```

```
g_Lcd.setCursor(8,1);                    g_Lcd.print("g");
g_Lcd.print(roundVal_3);                }
g_Lcd.setCursor(15,1);              }
```

Appendix B

Bill of Materials

Description	Qty	Unit	Unit Cost (Php)	Total Cost (Php)
120mm Fan	4	pcs	77	308
16x2 i2c LCD Screen	1	pcs	220	220
Adafruit Servo Driver	1	pcs	300	300
Buck Converter	2	pcs	75	150
Buttons	4	pcs	5	20
Desoldering Pump	1	pcs	74	74
DHT11 Temperature Sensor	2	pcs	125	250
Double Sided Circuit Board	1	pcs	50	50
Dual Channel Relay Module	4	pcs	75	300
Elbow PVC Pipe	1	pcs	60	60
ESP32	1	pcs	505	505
ESP32 Cam	1	pcs	429	429
Exhaust Fan	1	pcs	389	389
Fabrication and Labor	-	-	15000	15000
Filament	1	kg	850	850
Flat Cord	3	meters	30	90
Hinges	1	set	75	75
Hx711 Weight Sensor	4	pcs	70	280
Linear Actuator	1	pcs	1233	1233
Load Cell Bar	5	pcs	100	500
Mesh Screen	1	meter	120	120
Orange Pi	1	pcs	2000	2000

Plug	1	pcs	30	30
PVC Connector	5	pcs	40	200
PVC Pipes	5	pcs	106	530
RTC Module	1	pcs	50	50
Servo motor	4	pcs	250	1000
Superworms	4000	pcs	0.30	1200
Switching Power Supply	1	pcs	650	650
UTP Cable	13	meters	25	325
Vibration Motor	10	pcs	50	500
Webcam	1	pc	432	432
Wire #18	21	yards	5	105
Wire Organizers	1	set	48	48
Worm Gear DC Motor	1	pcs	470	470
TOTAL				28743

Appendix C

Specifications and Datasheets

Product	**Specifications**	**Brand**
Linear Actuator	1.Material: Aluminum alloy 2.Load capacity:72N 3.Speed: 30mm/s,15mm/s,9.5mm/s 4.Input Voltage: 12V DC 5.Duty cycle: 10%	Anggrek
Vibration Motor	Material: Iron Rated Voltage: DC 3.0V Working Voltage: DC 2.5~4.0V Working Temperature: -20'C ~ +60'C Rated Rotate Speed: Min. 9000RPM Rated Current: Max. 90mA Starting Current: Max. 120mA Starting Voltage: DC 2.3V Insulation Resistance: 10Mohm Terminal Impedance: 31ohm +/- 15% (single posture), 59ohm +/- 15% (double posture) Cable Length: 3cm Dimensions: 1.57 in x 0.39 in x 0.12 in (4.0 cm x 1.0 cm x 0.3 cm)	TinyTronics
DC Motor	Output Shaft: Eccentric Shaft Spin Direction: Bi-Direction Carbon Brush. Magnet: Rubber Magnet. Spur Gearbox. High Torque Power, Long Life-Span, Small Vibration. MIT / OEM / OBM. Professional DC motor / gear reducer customization in various speed, torques, voltages, output shafts, OD, etc. Specification Operating Voltage Range: DC 6V ~ 24V Gearbox OD: φ 30mm DC Motor OD: φ 29mm	Akali
Computer Fan	Brand: Ad-Link Fan Size: 12cm, and 8cm	Allan

	Connection: 4-Pin Molex Connector Interface: 4pin Molex Air Flow: 26.03CFM (+/-10%) Noise: 25dB (A) Bering Type: Oil Bearing Rate Voltage: 12VDC,0.1A Fan Speed: 2200RPM (+/-10%) Cable Length:30cm Weight:53G	
Exhaust Fan	Hanabishi Rust Proof Body. exhaust fan ventilation heavy duty. exhaust fan wall mounted for room. Hanabishi electric fan motor only. kitchen exhaust fan ventilation. exhaust fan ceiling mounted. INTAKE AND EXHAUST FEATURE. THERMAL FUSE PROTECTED MOTOR. INCLUDES METAL SHUTTER CONTROL. WALL MOUNTED. AVAILABLE IN 6, 8, 10, 12 BLADE	Futengbao
Rtc Module	Two Time-of-Day Alarms. Digital Temp Sensor Output. Register for Aging Trim. DS 3231 RTC with 2032 Battery Holder. Highly Accurate RTC Completely Manages All Timekeeping Functions. Real-Time Clock Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the Week, and Year, with Leap-Year Compensation Valid Up to 2100. Configurable I2C device Address for AT24C32 using SMD jumpers on PCB (A0, A1, A2). Programmable Square-Wave Output Signal. Battery-Backup Input for Continuous Timekeeping. Low Power Operation Extends Battery-Backup Run Time. Package Includes: 1 x DS3231 Precise Real-Time Clock Module Memory Module Operating Voltage (VDC) 2.7 ~ 5.5 Voltage Supply for RTC 2.2 V ~ 5.5 V Accuracy ±2ppm from 0°C to 40°C.	DS3231

	±3.5ppm from -40°C to 85°C. Battery Holder2032 Coin Battery. I2C interface Fast (400kHz) I2C Interface. EEPROM AT24C32 32Kbit Serial I2C. Time and Date Format Time: HH: MM: SS (12/24 hr). Date Format: YY-MM-DD-dd. Operating Temperature Range (°C) -40 to 85 Digital Temp Sensor Output 10 bit, ±3°C Accuracy and 0.25C resolution. Dimensions in mm (LxWxH) 38 x 22 x 14 Weight(gm) 8 Shipment Weight 0.012 kg Shipment Dimensions 12 × 8 × 5 cm	
Adafruit Servo Motor Driver	Feature: 1. It is convenient for you to insert 16 servo motors in 4 groups of 3-pin connectors at a time. 2. The plug of the servo motor is slightly wider than 0.1in, so you can put 4 pairs of 0.1in connectors. 3. Place a large capacitor on the V+ line, and the maximum external input voltage depends on this 10V1000uf capacitor. 4. There is a 220-ohm resistor on all PWM output lines to protect them and drive the LED easily. 5. 5V compliant, so a 3.3V microcontroller can control it and safely drive it up to 6V outputs. Specification: Item Type: PWM Servo Driver Product Material: Printed Circuit Board Model: PCA9685	PCA9685
Servo Motor	Operating Voltage: 4.8 ~ 6.6v Temperature Range: 0- 55℃ Standard Attachments Included Operating Speed: 0.13sec / 60 degrees (6v no load)	MG 95 Tower Pro
Esp 32	Operating Voltage: 2.3 ~ 3.6V. Operating Current: 80mA. Clock Frequency: 80 ~ 240MHz. Flash Memory: 4MB.	ESP32 Dev Kit Doit

Computer Webcam	Designed for both laptop and desktop. Built-in microphone, plug, and play. High definition and true color images. 360 degrees rotary function, all angle is available. Compatible with USB2.0/1.1. Auto white balance, auto color correction. Manual adjustable focal length. Support CC2000, AIM, Netmeeting, ICQ, MSN Messenger, Yahoo Messenger, and Skype, etc. Item color: Black Image sensor: CMOS Maximum dynamic pixel: 5.0M pixels Maximum static pixel: 50.0M pixels The focal length: 8 cm infinities Signal-to-noise ratio: More than 48 dB (30 F/s 220 LUX) WEIGHT 0.2 kg DIMENSIONS 7 × 16 × 15 cm	A4tech
Switching Power Supply	Input Voltage: 200 - 240 VAC; 100 - 120 VAC length: 15.9 cm width: 9.7cm height: 3.9cm weight: 425.5 g Frequency: 50/60 Hz Output Voltage: 12 VDC Output Current: up to 5 A	Max
Buck Converter	Input voltage :4V ~ 35V The output voltage: 1.23V ~ 30V Output Current :3A(Max) Conversion efficiency:92%(MAX) Output ripple: <30mV Switching frequency: 150KHz Working temperature: -forty Five~ +85 Size :43mm * 21mm * 14mm (Long*Width*Height)	LM2596s
16x2 I2C Lcd Screen	Specifications Is customized: Yes Model Number: LCD1602 Operating Temperature: 1 Dissipation Power: 1 Features for 1602 LCD:	Makerlab

	LCD display module with blue blacklight. Wide viewing angle and high contrast. Built-in industry standard HD44780 equivalent LCD controller. Commonly used in: copiers, fax machines, laser printers, industrial test equipment, networking equipment such as routers and storage devices. LCM type: Characters Can display 2-lines X 16-characters. Voltage: 5V DC. Module dimension: 80mm x 35mm x 11mm. Viewing area size: 64.5mm x 16mm Features IIC/I2C : Arduino 1602 I2C interface 4-wire 1602 screen IO port of Arduino control board is only 20,so IO ports is not enough for many sensor, SD card, relay modules. The original 1602 screen need 7 IO ports to drive up, and this module can save 5 IO ports	
Temp Sensor	Input Supply Voltage (VDC): 3.3 ~ 5. Supply Current (mA): measurement 0.3mA standby 60μA. Temperature measurement range: 0~50 degrees. Temperature measurement error: ±2 degrees. Humidity measurement range: 20%~95%RH. Humidity measurement error: ±5%RH.	DHT11
Weight Sensor	Type: weighing sensor set. Size: 12.7*12.7*80mm, 12.7*12.7*75mm. Color: as pictures show. Weight: 1kg, 5k, 10kg, 20kg.	Hx711
Load Cell Bar	Model: YZC-131 Material: Aluminum Dimensions: 80x13x12.7mm Range: 20kg Rated output 1.0 ± 0.1mV / V Operating temperature range -20 ~ 65°C Weight:32g	Makerlab
Orange Pi	Features: 1.CPU: H3 Quad-core Cortex-A7 H.265/HEVC 4K 2.GPU:Mali400MP2 GPU @600MHz·Supports OpenGL ES 2.0 3.Memory: 512MB DDR3 (shared with GPU)	Makerlab

	4.Onboard Storage: TF card (Max. 64GB) / MMC card slot 5.Onboard Network: 10/100M Ethernet RJ45 6.Video input: A CSI input connector Camera: Supports 8-bit YUV422 CMOS sensor interface Supports CCIR656 protocol for NTSC and PAL Supports SM pixel camera sensor Supports video capture solution up to 1080p@30fps 7.Video output: Supports HDMI output with HDCP Supports HDMI CEC Supports HDMI 30 function Supports simultaneous output of HDMI 8.Power: DC input can supply power, but USB OTG input don't supply power 9.USB 2.0 ports: Only One USB 2.0 HOST, one USB 2.0 OTG 10.Button:Power Button(SW4) 11.Low-level peripherals: 40 Pins Header,compatible with Raspberry Pi B 12.LED lights: Power led & Status led 13.Key:POWER 14.Supported operating systems : Android Ubuntu, Debian, Rasberry Pi Image 15.Size:69 mm × 48mm 16.Weight:36g	

Appendix D

Project Manual

SOFTWARE

OVERVIEW

The mobile application provides features that are useful for keeping track of the Superworms. The features are as follows:

1. Notification System – The sensors of the system will send the inputs to the application, allowing the user to access the information about the weight and temperature.
2. Time Graph – The time graph in the mobile application is updated each time new data is received.
3. Live Video of Superworms – Monitoring and counting process is made more efficient and accurate.

Home Page

1. Home page displays the data that the system measures and records such as temperature, humidity, weight of the containers, controls, time graph and computer vision.

SSID and Password

1. Click the  icon located at the upper left corner of the Home Page.
2. Input the network ID and password of the Wi-Fi that you will be connecting to.
3. Click Submit.

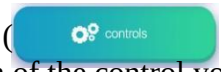
Live Video of Superworms

1. Click the superworms button (

).
2. Choose the container you want to see.
3. Click the back button in the upper left corner to go back to the options of containers.
4. Click the back button 

 again if you want to go back to the Home Page.

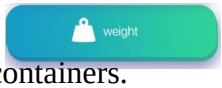
Controls

1. Click the controls button (

).
2. Click the ON/OFF button of the control you want to open/close.
3. If you want to go to the Food Dispenser section, click the Food Dispenser button in the lower right part of the page.
4. Click the ON/OFF button of the level you want to open/close.
5. Click the Controls button in the lower left corner if you want to go back to the
6. Controls section. If you want to go back to the home page, click the 

 button.

Weight of the Containers

1. Click the weight button (

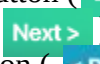
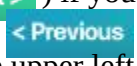
).
2. Check the weight of the containers.
3. Click the back button 

 if you want to go back to the Home Page.

Refresh

1. Click the refresh button () if you want to refresh the data in the application.

Time Graph

1. Click the time graph button () if you want to see the time graph.
2. Click the next button () if you want to see the other graphs.
3. Click the previous button () to go back to the first page of graphs.
4. Click the back button at the upper left corner if you want to go back to the Home Page.

HARDWARE

1. Put the Worms in the 1st container
2. Put the Plastics in the funnel
3. Plug in the Machine

Appendix E

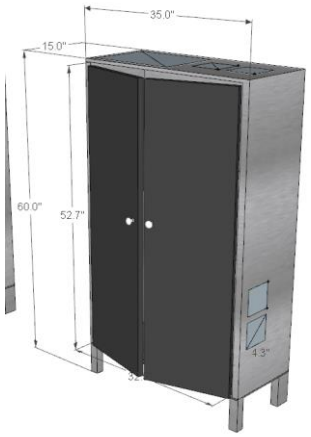
Duplication Manual

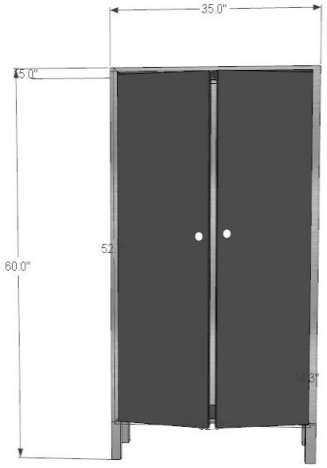
Hardware ***Duplication:***

A. Design and 3D modelling

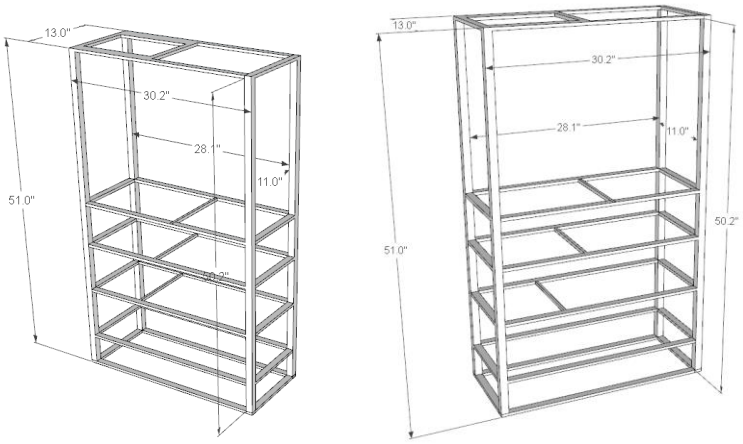
a. Enclosure and Skeletal Frame

For the design of the Skeletal Frame and Enclosure, the researchers used a free 3D-computer aided design tool called Sketchup to model the entire enclosure and Skeletal Frame. The researchers will use aluminum steel as the main material. By using a steel cutter and a welding machine, the dimensions of the enclosure will be created. The researchers will also employ a drawer mechanism where each compartment can be removed with ease. This will be done by using a hinge rail for each compartment. The enclosure of the system will ensure the safety and proper storage of the organisms. This will also help in preventing pests from entering and damaging the habitat of the organisms.





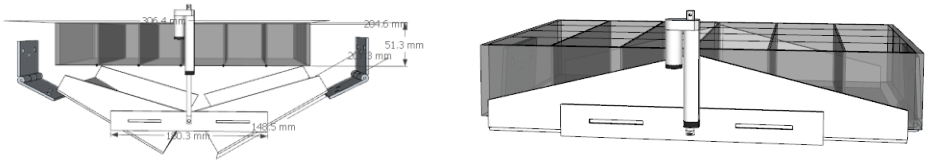
Enclosure Design



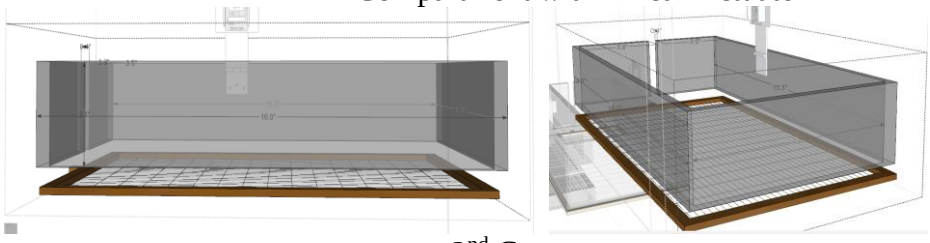
Skeletal Design

b. Mesh Screen and Compartments

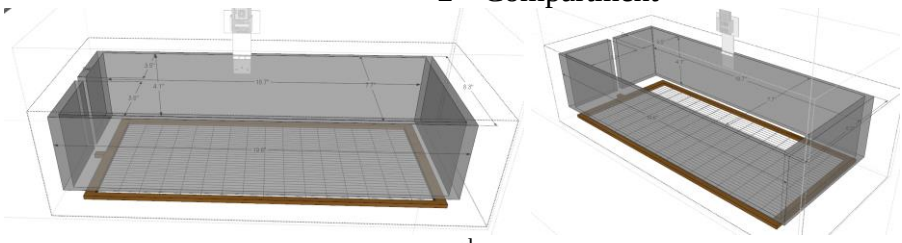
In designing the compartments, the mechanisms to be used got considered so that the process of rearing became feasible. The sizes of mesh screens were adjusted to the needed size in the actual design.



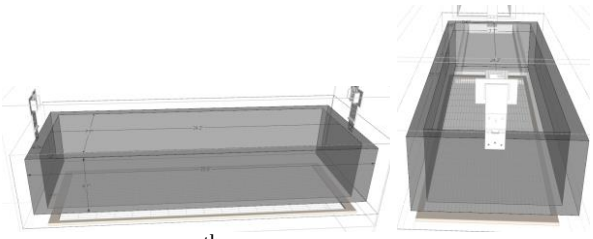
1st Compartment with Linear Actuator



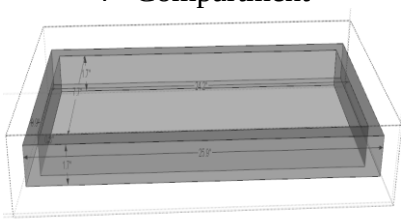
2nd Compartment



3rd Compartment



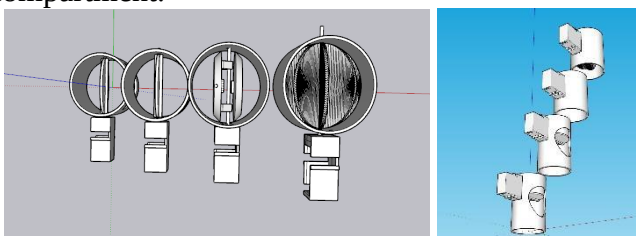
4th Compartment



Frass Compartment

c. Food Dispenser Tubes

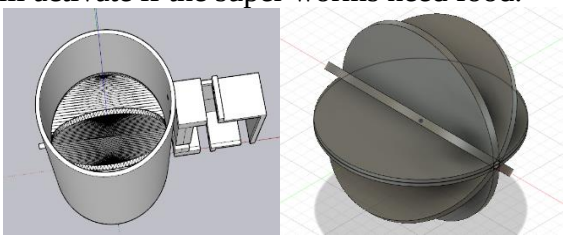
For the Food dispenser, the researchers created a faucet like mechanism using servo motors and joint tubes to act as a food distributor. The food processing system consists of 4 different levels which have its own function. The creation of the mechanisms namely the Tube Switch, Stopper and the Weight switch is discussed in this part. By using this mechanism, the researchers will be allowed to distribute a specific amount of plastic for a desired compartment.



4 Parts of the Food Dispenser Tube

1. Stopper Mechanism

   This part of the system will act as the stopper and the main distributor of food in the food processing system. The servo motor is programmed such that it will activate if the super worms need food.

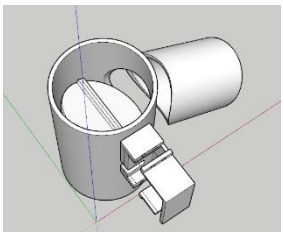


Stopper Mechanism

2. Tube Switch Mechanism

   By utilizing the rotating motion of the servo motor, a directional based switch can be created by allowing the plastics to free fall and slide onto the tube with will act as the switching mechanism of the Food

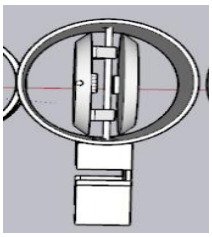
Mechanism. The code of the servo motor was set to activate to 90 degrees if a certain parameter is achieved and will return to its former state if not.



Tube Switch Mechanism

3. Weight Switch Mechanism

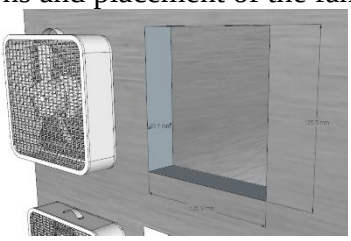
The weight switch mechanism uses a loadcell and a servo motor as its key functionality. When a certain amount of weight is measured by the loadcell, the servo motor will rotate 90 degrees and after some time the servo motor will go back to the original state. An interrupt is used in this mechanism so that when the servo motor moves 90 degrees, the measurement of the load cell will stop and resume only when the servo motor goes back to its normal state.



Weight Switch Mechanism

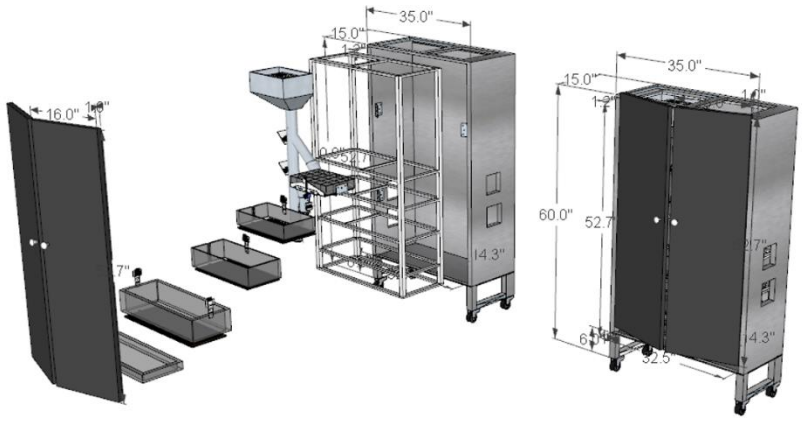
d. Ventilation System

The ventilation system will consist of 6 120x25 mm fans, 4 for the intake placed on each side of the enclosure, 2 for the exhaust placed on the top of the enclosure. This is to ensure that the optimal airflow is achieved in the system. The dimensions and placement of the fans are shown below.



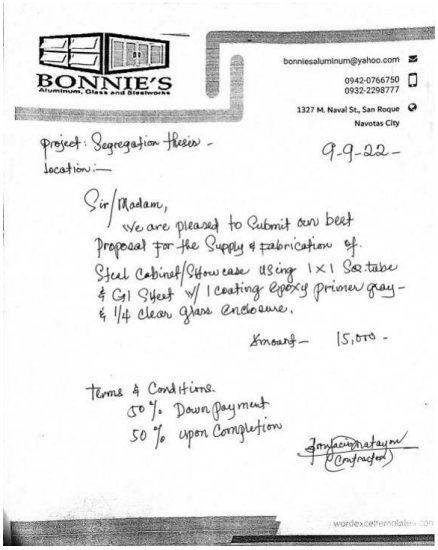
Fan Placement Hole

e. Whole Exploded Design



B. Fabrication of Enclosure, Skeletal Frame, Compartments and Mesh Screen

For the Fabrication of all the designs, the researchers contracted Bonnie's Aluminum Steel and Glass manufacturer to fabricate the designs created by the proponents. The fabricator is located around Navotas City, the receipt is attached in the image below.


BONNIE'S
Aluminum, Glass and Steelworks

bonniesaluminum@yahoo.com
0942-0766750
0932-2298777
1327 M. Naval St., San Roque
Navotas City

Project: Segregation Fleeis -
Location: —

9-9-22 -

Sir/Madam,
We are pleased to Submit our best Proposal for the Supply & Fabrication of Steel Cabinet/Showcase using 1x1 Sq. tube & GI Sheet w/ 1 coating Epoxy primer gray - & 1/4 clear glass enclosure.

Amount — 15,000 -

Terms & Conditions
50 % Down Payment
50 % upon Completion

Thank you
Contractor

Final Quotation of the Fabrication

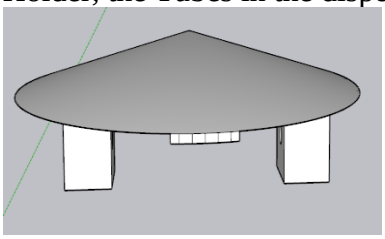
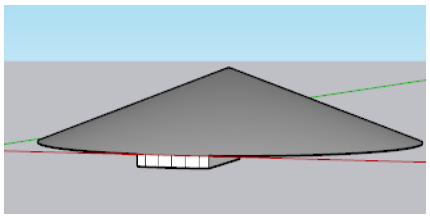

Images During the Fabrication of the Enclosure and Skeletal Frame

Finished Fabricated Skeletal Frame and Enclosure

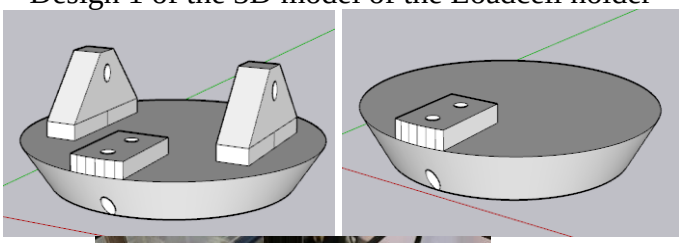
During the Fabrication process, one of the members always visit the fabrication site to supervise the fabricator and check for adjustments made.

C. 3D Printing of Food Dispenser Tubes, Loadcell Placeholder

For the 3D printing of the tubes and Loadcell Placeholder, there were several attempts to perfect the quality of the 3D printed design. Since the most challenging piece of the 3D printing process is the Loadcell holder for Weight Switching Mechanism, the users iterated the 3D printing process and design 2 times to achieve a working model that can measure weight at the same time switch with the Servo motor. The following are the images of the trials in 3D printing of the Loadcell Holder, the Tubes in the dispenser and the servo-motor holder.

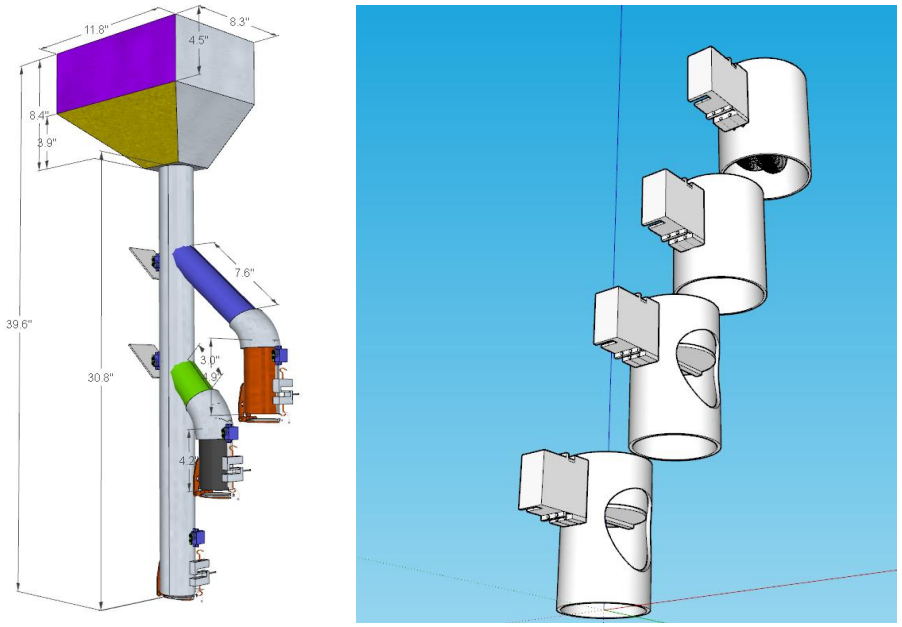



Design 1 of the 3D model of the Loadcell holder



Design 2 of the Loadcell Holder

For the Servo motor holder and the angled tubes, the lengths of each tube were measured according to the placement of each compartment. Here are the following images regarding the measurements.



Design and Measurement of the Food Dispenser Tube

3D printing of the Servo Motor Holder, Angled Tubes, Tube Switch and Weight Switch

D. Procurement of Materials

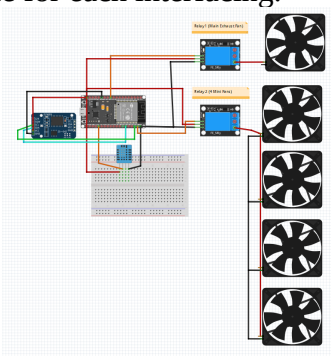
For the procurement of materials, the researchers used any mediums available to acquire the materials. The researchers utilized online shops such as Lazada and Shopee to order the parts required while considering the specifications and parameters of each device/sensor. Here is the list of the procured parts for the system.

E. Interfacing of different Sensors and Devices

In the interfacing of different sensors, each has been grouped according to its system. Every sensor and mechanism used in this system has its own purpose. Each interfacing will be discussed below.

a. Ventilation System

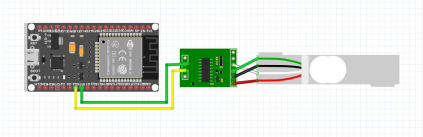
For the ventilation system, there are 4 devices/sensors used. The Ventilation system uses 1 DHT11 humidity and temperature sensor, 1 RTC module (temperature measurement), DC Fans and Relays. Using an Arduino IDE and an ESP 32 as the main microcontroller, the researchers used source codes for the DHT11, the RTC module and the relays to set the living conditions required for the system. The researchers set a temperature-based switch for the Fans so that, when a specific temperature is met, the Fans will turn ON. Here is the code for each interfacing.



Circuit of the DHT11, RTC Module and Fans connected to 2 relays.

b. Weight System

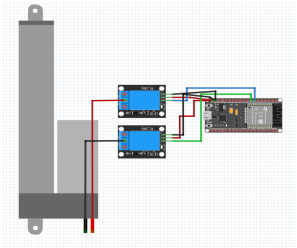
For the weight system, the researchers used a HX711 Loadcell Amplifier, Loadcell Bars, and an ESP-32 as the main microcontroller. The basic templates of the HX711_ADC sketches were used and modified to measure the weight and calibrate. The researchers explored the library of HX711_ADC to unlock the full potential of the weight sensor. Here are the codes used to interface the weight sensor and the circuit used.



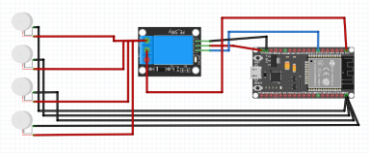
HX711 Loadcell Circuit with ESP-32

c. Segregation System

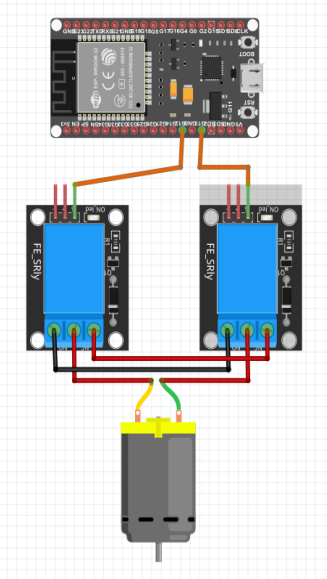
For the segregation system, the mainly used modules here are relays because the devices used in each segregation functions are DC motors, Linear actuators and Vibration Motors which all requires a Switching from On to Off. Here are the codes used to operate the switching of the motors that are connected already to the database.



Linear Actuator and Relay Circuit with ESP-32



Vibration motor and Relay Circuit with ESP32

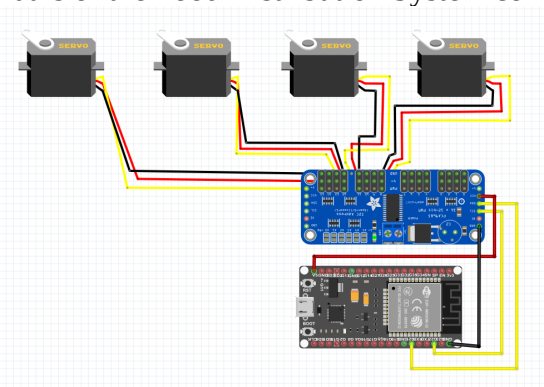


DC Motor pulley with Relay Circuit with ESP-32

d. Food Dispenser System

The Food Dispenser System has multiple devices in it namely 4 Servo Motors, 1 Loadcell Bar, 1 HX711 Amplifier, 1 Adafruit Servo Motor Driver and the 3D printed parts. The interfacing of the food dispenser is considered the most challenging because it has the weight switch mechanism that always recalibrate while the servo motor rotates. Here are the codes used to interface the Food dispenser and the algorithm to change the food path distribution.

Switch Paths of the Food Distribution System connected to the Database.



Servo Motor, Adafruit Servo Motor Driver Circuit Diagram with ESP-32

e. Control Panel

For the control panel, the devices used are a 16x2 I2C LCD, 4 buttons and an ESP-32. MBLIB library was used in this control panel. This was the easiest Control Panel sketch available to modify for the needs of the system. Here is the code used for the Control Panel.

F. Assembly of the System

a. Assembly of the Main Circuit to the Board

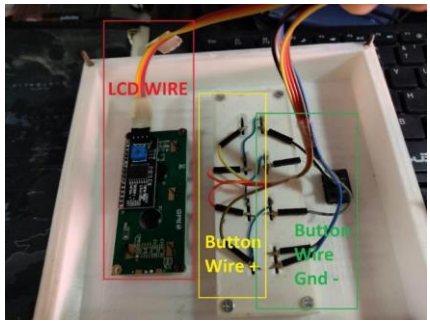
1. Mount the Board, Power supply and the Control Panel to the placement area.
2. Use appropriate Screws and tighten the nuts.
3. Connect the Power wires of the power supply to the supply of the main circuit board.

b. Assembly of the Control Panel and its Connection to Mainboard.

1. The Control Panel has 3 main wires that connects to the Board.

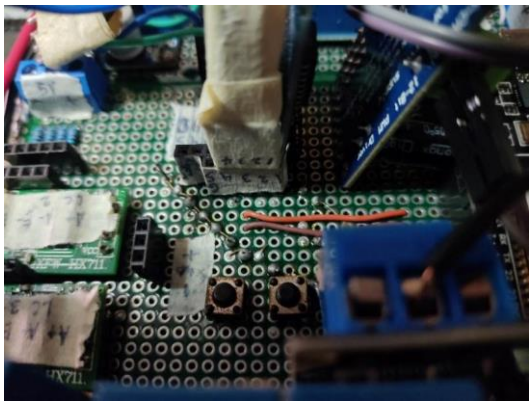
Control Panel Wire Part	# of Wires	Color of Wires

LCD Wire	4	Green, Blue, Violet and White
Button Wire (+)	4	White, Gray, Violet and Blue
Button Wire GND (-)	5	Black, Brown, Red, Orange and Yellow

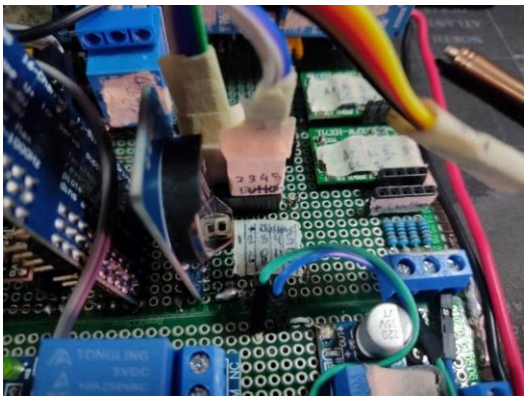


2. Follow the Specific Connections of the Control Panel. ***Note: STRICTLY follow the connection because it might short the LCD or the ESP-32.***

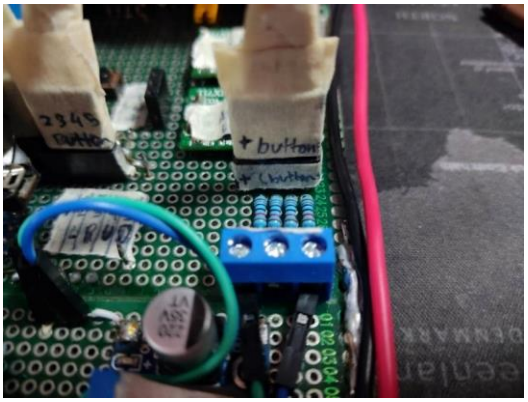
i. LCD Wire



ii. Button Wire (+)

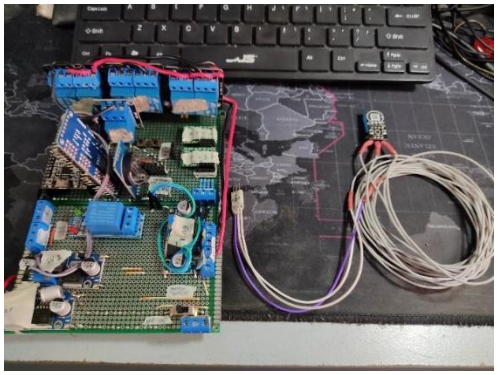


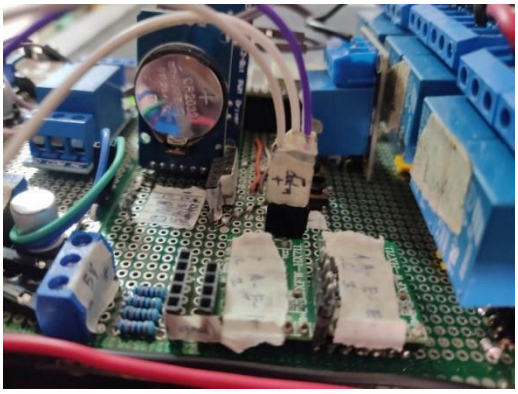
iii. Button Wire GND (-)



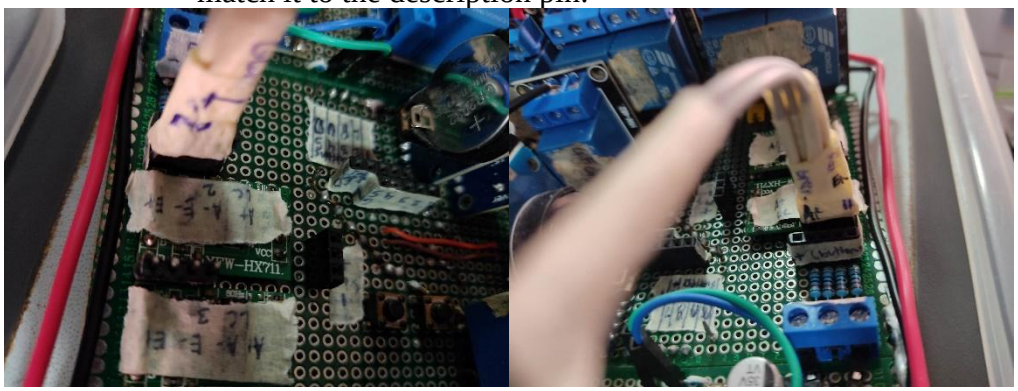
iv. The assumed operation of the control panel if the connection is right is here:

c. Connection of extended DHT11 sensor.





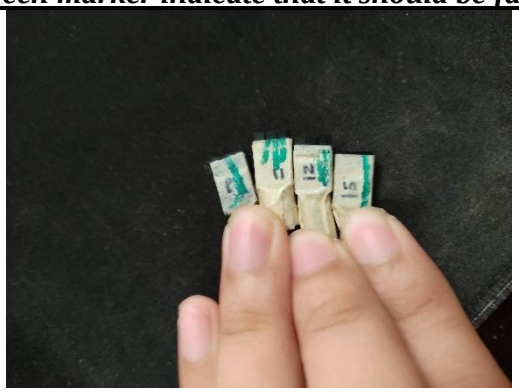
d. Connection of Loadcell 2 and 3
    1. Follow the proper connections of E+ and A+ pins.
    2. For loadcell 2 here is the connection, Notice the E+ and A+ label and match it to the description pin:

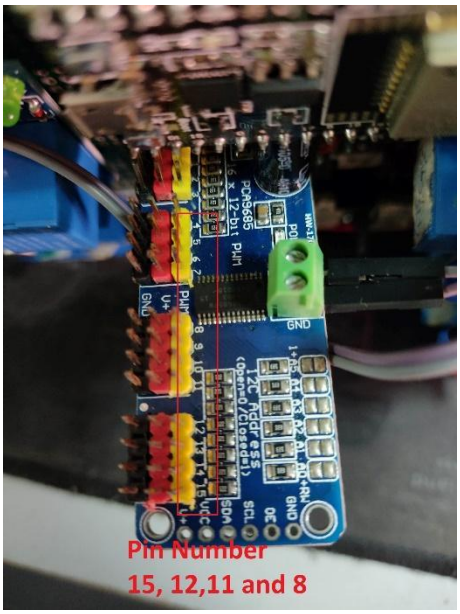


    3. For loadcell 3, it is yet to be finished. But the same connection is applied.

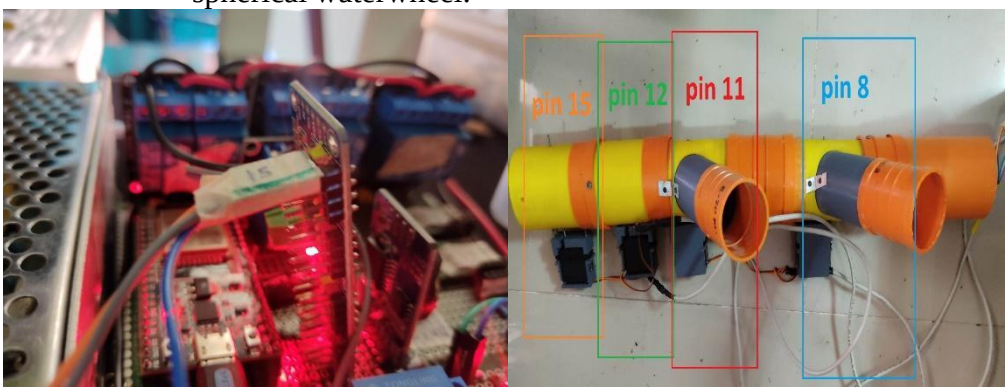
e. Connection of Food Dispenser Tube to the Main Circuit Board
    1. Install the Food Dispenser Tube first to the placement area.
    2. Place the appropriate pin that corresponds to the pin number of the PWM servo motor driver.

Note: The Green marker indicate that it should be faced forward

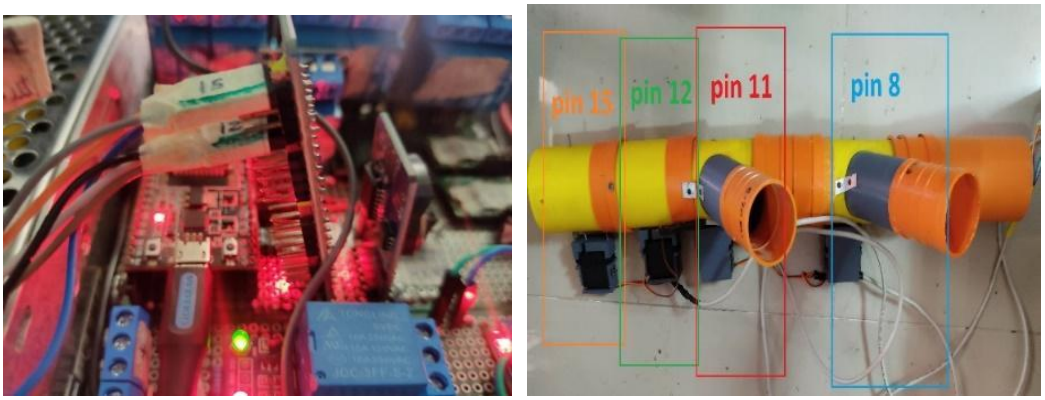




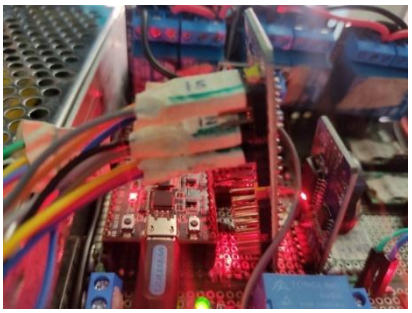
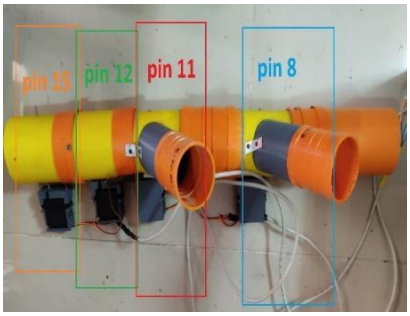
3. Pin 15 corresponds to the first part of the tube which consist of the spherical waterwheel.



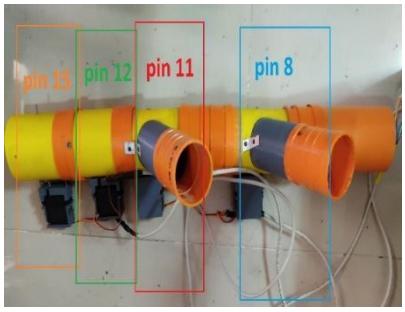
4. Pin 12 corresponds to the 2nd part of the tube which uses a loadcell as a swivel.



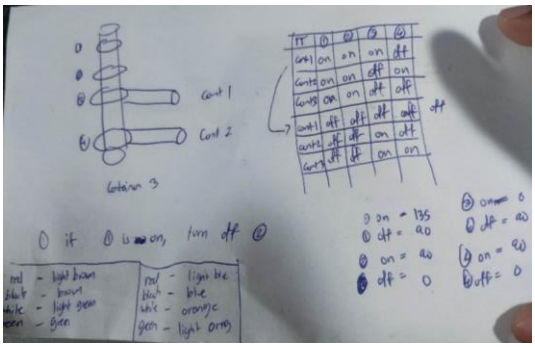
5. Pin 11 corresponds to the 3rd part of the tube which uses a plastic as a swivel.

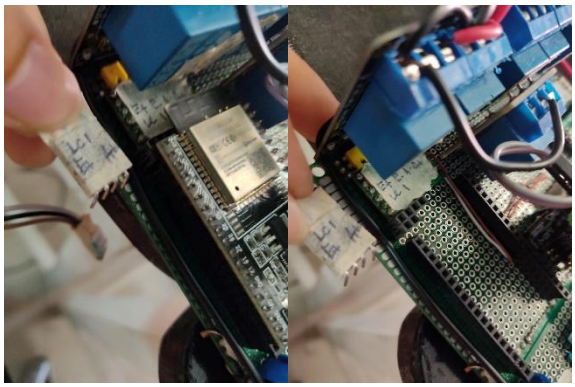
6. Pin 8 corresponds to the 4th part of the tube which uses a plastic as a swivel.

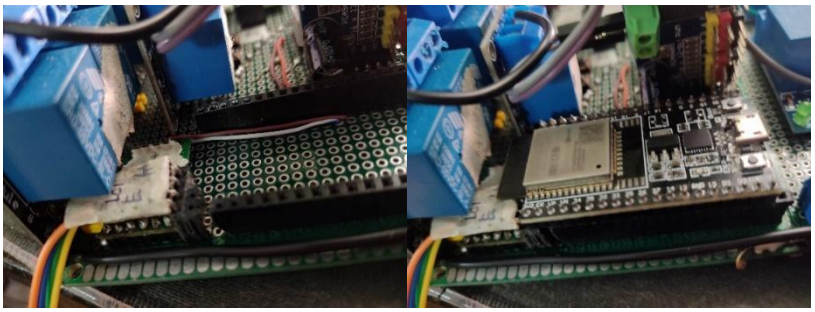
7. The expected output of the Tube Mechanism. Loadcell Swivel (2), Switch 1 (3) and Switch 2 (4) has a set of instructions that will help send the food to its own container.



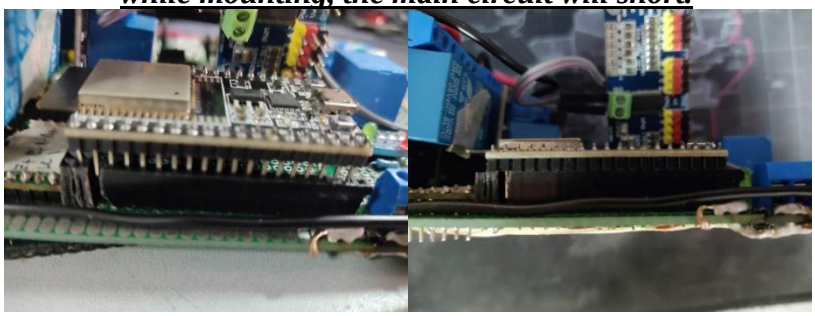
8. For the Loadcell 1 that is inside the 2nd part of the tube, there is a specific step to connect it to the board. The Loadcell 1 female pin is located at the top of the ESP32. Here are the steps to mount the Loadcell 1.
9. First Remove the ESP-32



10. Mount the male pin to the female pin (follow the label of E+ and A+) according to the image shown below and mount back the ESP-32.

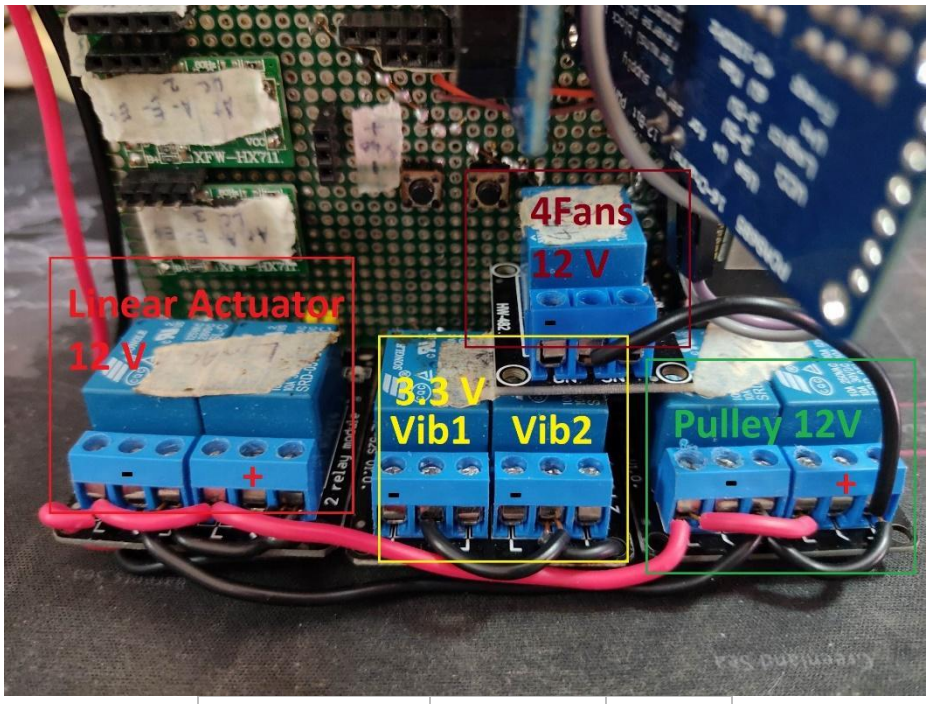


Note: Always check the mounting of the esp32 to the board, the following images are the proper mounting of the board. If the board is not mounted properly or some pins are interchanged while mounting, the main circuit will short.



f. Connection of Relays

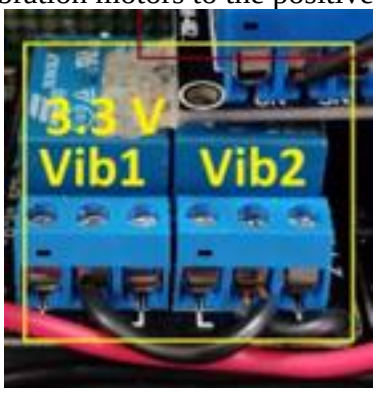
The circuit have a total of 8 relays. Each having its own functions. Follow the proper connection of the relays. Troubleshoot if necessary.

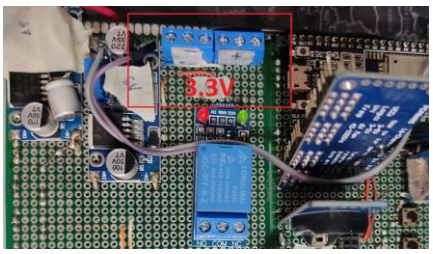


Relay Name	# Number Relay	Voltage
Linear Actuator	2 Relays	12V
Vibration motor 1 and 2	2 Relays	3.3V
Pulley	2 Relays	12V
4 Fans	1 Relay	12V
Exhaust Fan	1 Relay	

1. Linear Actuator Connection
   Follow the connection, reverse the polarity depending on the reaction of on and off the circuit.

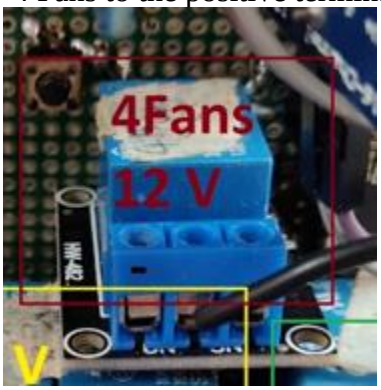
2. Vibration Motor 1 and 2
   Follow the connection, the connection only consists of a negative terminal. Connect the negative terminal of the vibration motor of the 2nd container to Vib1 and the 3rd container to Vib2. Then connect both positive terminals of the vibration motors to the positive terminal of 3.3V.

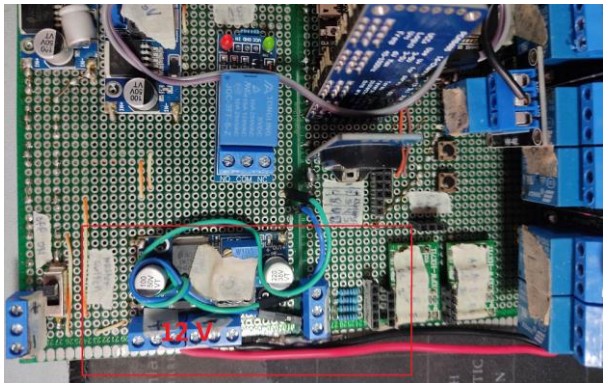




3. Pulley
   Follow the connection, reverse the polarity depending on the reaction of on and off the circuit. Put the negative and positive terminal of the pulley as seen in the image below.

4. 4 Fans
   Follow the connection, the connection only consists of a negative terminal. Connect the negative terminal of the 4 Fans. Then the positive terminals of the 4 Fans to the positive terminal of 12V.





.

FRONT-END DEVELOPMENT

. Development of the User Interface of the Application

1. The first step is to use Adobe Photoshop to construct the wireframe. Wireframes are used to build the basic structure of a page before visual design and content is added.
2. After setting the foundation for the user interface, Adobe Photoshop and Canva are used to design the various components, including the slides, buttons, and icons.
3. After constructing the design, the components and layouts will be uploaded in the assets of the Thunkable.
4. Using the editing features in Thunkable, the components and layouts can now be used.

BACK-END DEVELOPMENT

. DEVELOPMENT OF THE BACK-END OF THE MOBILE APPLICATION

1. Set the api key and database url of the realtime database.
2. 2. In the Blocks section of Thunkable, here is where you can create blocks that serve as the backend of the mobile app.
3. 3. Create a cloud variable with the same name as the key in your realtime database.
4. 4. You can now read and write to your realtime database using the different blocks in Thunkable.

B. DATA VISUALIZATION OF THE PROJECT

1. The time graph will be developed by using Google Sheets and Google Firebase.PPS
2. The Google Apps Script of Google Sheets will be used for programming the codes that will be able to run and sync the data from the Firebase to Google Sheets.
3. Everytime the Firebase receives new data, it will automatically update the Google Sheets and display it in a graph form.
4. To view the graphed data in Thunkable, a web viewer will be used to link the graphs from Google Sheet

Appendix F

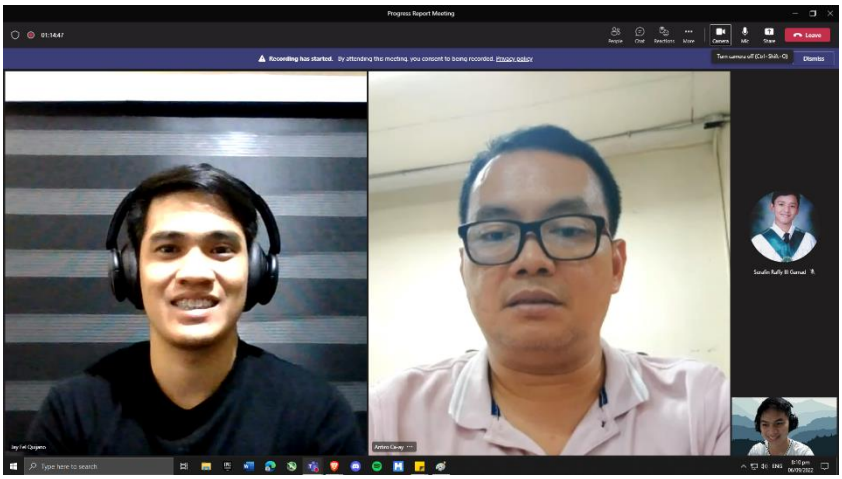
Project Documentation

Project Making

3D Printing

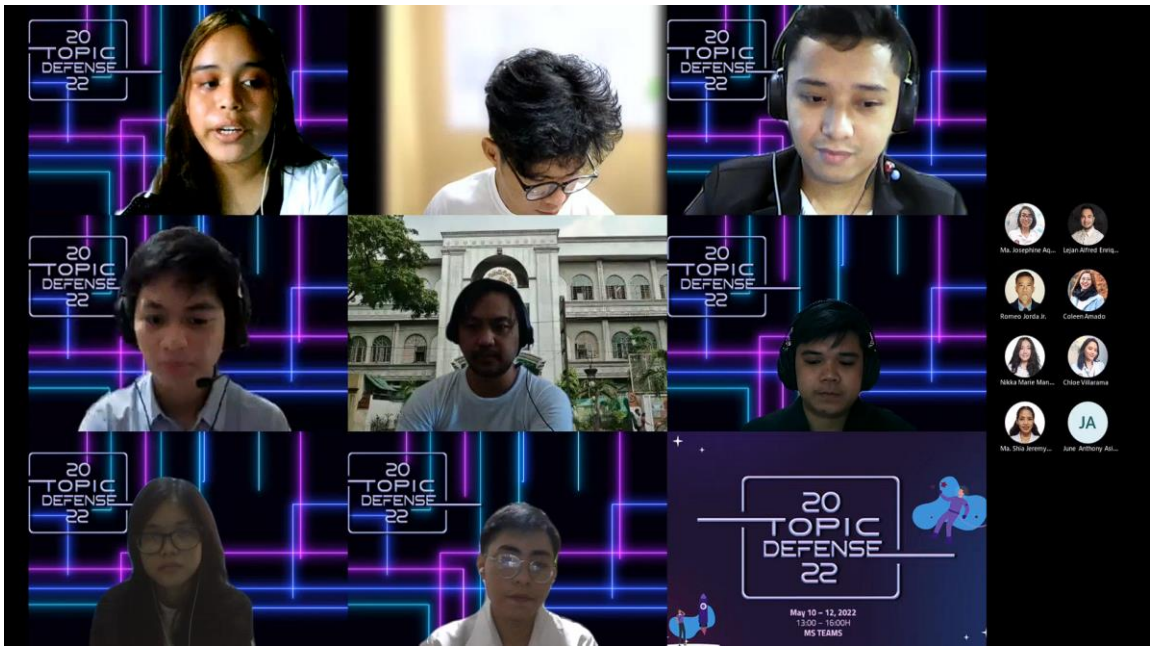
Site Visit

Business meeting with sir Lendl



Consultation meeting

Topic Defense

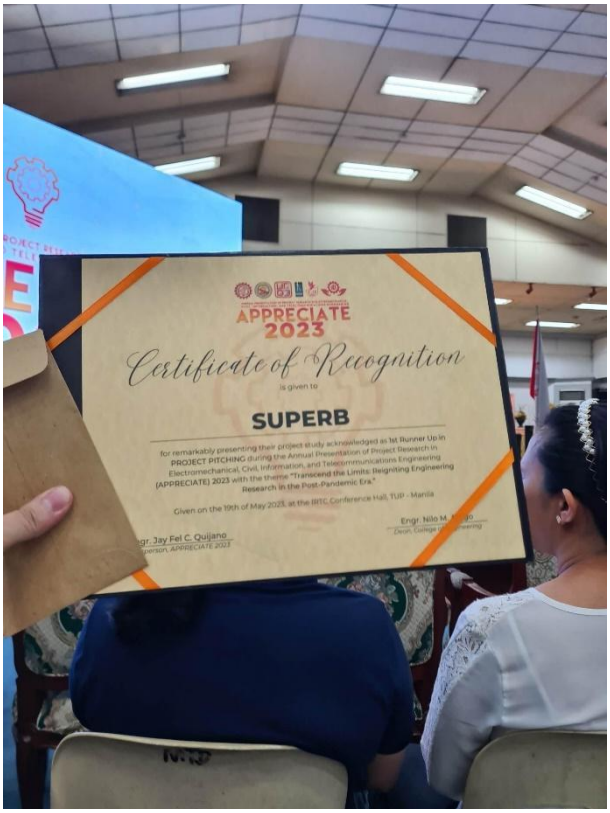


Title Defense

Progress Defense

Pre Final defense


SUPERB
METHODOLOGY
METHODOLOGY


APPRECIATE
2023
Certificate of Recognition
is given to
SUPERB
for remarkably presenting their project study acknowledged as 1st Runner Up in
PROJECT PITCHING during the Annual Presentation of Project Research in
Electromechanical, Civil, Information, and Telecommunications Engineering
(APPRECIATE) 2023 with the theme "Transcend the Limits: Reigniting Engineering
Research in the Post-Pandemic Era."
Given on the 19th of May 2023, at the BITC Conference Hall, TUP - Manila
Engr. Jay Fel C. Quijano

Appreciate 2023

Final Defense

Appendix G

Researcher's Profile

JOHN CEDRIC M. ANGELES

RESEARCHER

09271832479

johncedric.angeles@tup.edu.ph

811 Centro St. Sampaloc, Manila

linkedin.com/in/jcedangeles/

PROFILE

A graduating BS Electronics Engineering student with proficient skill sets in the software development in various languages. Seeking opportunities to apply skills and contribute to innovative projects in the field.

SKILLS

- Software Development
- Critical Thinking
- Problem-Solving
- Computer Literacy
- Project Management Tools
- Strong Communication

EDUCATION

TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES

BS Electronics Engineering

2019-2023

POLYTECHNIC UNIVERSITY OF THE PHILIPPINES

Science, Technology, Engineering, Mathematics (SHS)

2017-2019

UNIVERSITY OF THE EAST

Junior High School

2013-2017

CO-CURRICULARS

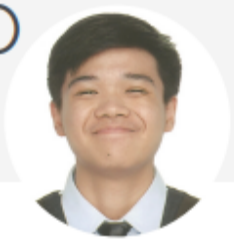
CHIEF INNOVATIONS OFFICER

Google Developer Student Club - TUP

2022-2023

CARLO E. BARRETTO

RESEARCHER



09669503141

barretto.carlo.e@gmail.com

Cynthia Nakpil St., BFRV, Talon Dos, Las pinas City

PROFILE

A graduating Electronics Engineering student with proficient software development skills. Seeking opportunities to apply skills and contribute to innovative projects in the field, particularly those that use emerging technologies to solve real-world problems.

SKILLS

- Problem Solving Skills
- Attention to Detail
- Python Language
- Problem-Solving
- Computer Literacy
- Project Management Tools
- Strong Communication

EDUCATION

TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES - MANILA
BS - ELECTRONICS ENGINEERING
2019 - 2023

WESTFIELD SCIENCE ORIENTED SCHOOL
Junior High School to Senior High School (STEM)
2014 - 2016

ELIZABETH SETON SCHOOL - MAIN CAMPUS
Primary Education
2007 - 2013

ORGANIZATIONAL AFFILIATIONS

ORGANIZATION OF ELECTRONICS ENGINEERING STUDENTS
Member
2019 - 2023

IINSTITUTE OF ELECTRONICS ENGINEERS OF THE PHILIPPINES - MANILA STUDENT CHAPTER
Member
2019 - 2023

SERAFIN RAFFY III C. GAMAD

RESEARCHER

SERAFIN "RAFFY" III C. GAMAD

09994940892

serafingamad1234@gmail.com

Bldg4 Unit117 Campville Cupang Muntinlupa

serafingamadiii.jobs180.com

PROFILE

I am an Arduino hobbyist, hardware developer, and skilled 3D designer. With a deep passion for music and its seamless integration with technology, I strive to create innovative projects that bridge the two worlds. Beyond my technical pursuits, I am driven by a constant desire to explore and discover the beauty found in life's smallest and most delightful moments.

SKILLS

- 3D Modelling
- Arduino Coding
- Python Coding
- Sensors Interfacing
- Problem-Solving
- Computer Literacy
- Project Management Tools
- Strong Communication Skills

EDUCATION

TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES

BS Electronics Engineering

2019-2023

MUNTINLUPA SCIENCE HIGH SCHOOL

Science, Technology, Engineering, Mathematics (Senior High School)

2017-2019

MUNTINLUPA SCIENCE HIGH SCHOOL

Junior High School

2013-2017

ORGANIZATIONAL AFFILIATIONS

ORGANIZATION OF ELECTRONICS ENGINEERING STUDENTS

Member

2019 - 2023

INSTITUTE OF ELECTRONICS ENGINEERS OF THE PHILIPPINES - MANILA STUDENT CHAPTER

Member

2019 - 2023

ASSOCIATE TECHNICIAN

STELSEN INTEGRATED SYSTEMS

2022 August-2022 September

- Electrical Management ,Repair and Mapping of FDAS (Fire Detection and Alarm System)

HARDWARE DEVELOPMENT AND ARDUINO INTERFACING

SUPERB (Thesis at Technological University of the Philippines Manila

2022-Present

- Managed hardware development and device operations
- Implemented circuit development and management strategies
- Developed and interfaced with various sensors through Arduino prototyping and coding

LEANDRO C. MANLUTAC

RESEARCHER

09982730513/09498990331

lcmanlutac9.1@gmail.com

704 Blk.1 Dubai Brgy. 649, Baseco Port Area, Manila

www.linkedin.com/in/leandro-manlutac-a3497027b

PROFILE

I am a graduating student under Bachelor of Science in Electronics Engineering. I am committed to developing a deep understanding of the field and acquiring practical skills to contribute to the advancement of modern society. Looking to leverage my knowledge and skills into a role as an Electronics Engineering student.

SKILLS

- Basic Programming (Octave, MATLAB)
- 3D Modelling
- Committed to providing excellent quality work
- Problem-Solving
- Computer Literacy
- Adept and open to constructive criticism
- Strong Communication
- Proven leadership skills

EDUCATION

TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES

BS Electronics Engineering

2019-2023

ARELLANO UNIVERSITY - JUAN SUMULONG CAMPUS

Science, Technology, Engineering, and Mathematics (SHS)

2017-2019

JOSE ABAD SANTOS HIGH SCHOOL

Junior High School

2013-2017

ACHIEVEMENT/EXTRA CURRICULAR

- **DOST SCHOLAR | R.A. 7687 | 2019 - PRESENT**

MEMBERSHIP AND AFFILIATION

YOUTH FOR BETTER BASECO (YBB)

NGO | Member | March 2022 - Present

INSTITUTE OF ELECTRONICS ENGINEERS OF THE PHILIPPINES MANILA

University Organization | Member | 2019 - 2023

ORGANIZATION OF ELECTRONICS ENGINEERING STUDENTS

University Organization | Member | 2019 - 2023

DOST CLUB

University Organization | Member | 2019 - 2023

AIRA MARI W. MAYUNO

RESEARCHER

0943-058-9957

mayunoaira23@gmail.com

22 Gervacio St. Hulong Duhat, Malabon City

www.linkedin.com/in/airamarimayuno/

PROFILE

An ambitious, hardworking and dedicated graduating student who is passionate about the field of electronics and communication. Searching for an opportunity to enhance and contribute my skills and knowledge.

SKILLS

- Web Design
- Design Thinking
- Wireframe Creation
- Front End Coding
- 3D Modelling
- Problem-Solving
- Computer Literacy
- Adaptibility
- Strong Communication
- Ability to work independently or with groups

EDUCATION

TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES

BS Electronics Engineering

2019-2023

MALABON NATIONAL HIGH SCHOOL

Science, Technology, Engineering, Mathematics (Senior High School)

2017-2019

MALABON NATIONAL HIGH SCHOOL

Enginnering and Science Program (Junior High School)

2013-2017

ORGANIZATIONAL AFFILIATIONS

ORGANIZATION OF ELECTRONICS ENGINEERING STUDENTS

Member

2019 - 2023

INSTITUTE OF ELECTRONICS ENGINEERS OF THE PHILIPPINES - MANILA STUDENT CHAPTER

Member

2019 - 2023

EXPERIENCE

INQUIRER INTERACTIVE, INC.

Information Technology Intern

August - September (2022)

PUBLIC EMPLOYMENT SERVICE OFFICE (MALABON CITY)

Special Program for Employment of Students (SPES)

May (2016, 2017 and 2018)

Appendix H

Turnitin Results

turnitin

Similarity Report ID: oid:25992:38621562

17% Overall Similarity

Top sources found in the following databases:

- 12% Internet database
- Crossref database
- 12% Submitted Works database
- 7% Publications database
- Crossref Posted Content database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	academic.oup.com Internet	<1%
2	robu.in Internet	<1%
3	researchgate.net Internet	<1%
4	orangepi.org Internet	<1%
5	uctronics.com Internet	<1%
6	Submitted on 1688550709967 Submitted works	<1%
7	geografi.unima.ac.id Internet	<1%
8	la.mathworks.com Internet	<1%

Sources overview

9	Technological University Of The Philippines on 2019-03-20 Submitted works	<1%
10	polyseg.online Internet	<1%
11	Submitted on 1688008857098 Submitted works	<1%
12	dx.com Internet	<1%
13	amazon.com Internet	<1%
14	dokumen.pub Internet	<1%
15	researchspace.auckland.ac.nz Internet	<1%
16	camera.localbusinessbooster.info Internet	<1%
17	sweesengrg.com.sg Internet	<1%
18	mdpi.com Internet	<1%
19	firebase.google.com Internet	<1%
20	hsiangneng.com Internet	<1%

Sources overview

21	esp.org Internet	<1%
22	Shu-Cai Li, Bang Li, Xiao-Mei Qin. "Properties of LLDPE Film Modified b... Crossref	<1%
23	Anja Malawi Brandon, Shu-Hong Gao, Renmao Tian, Daliang Ning, Sha... Crossref	<1%
24	University of Technology on 2023-02-01 Submitted works	<1%
25	Bjørn Petter Jelle. "Nano-based thermal insulation for energy-efficient ... Crossref	<1%
26	store.roboticsbd.com Internet	<1%
27	coursehero.com Internet	<1%
28	makerlab-electronics.com Internet	<1%
29	robotaki.com Internet	<1%
30	elibrary.buse.ac.zw:8080 Internet	<1%
31	Submitted on 1686181630977 Submitted works	<1%
32	Glasgow Caledonian University on 2022-11-17 Submitted works	<1%

Sources overview

33	Macquarie University on 2021-05-10 Submitted works	<1%
34	maxapress.com Internet	<1%
35	Loudoun County Public Schools on 2022-09-07 Submitted works	<1%
36	Angeles University Foundation on 2023-03-16 Submitted works	<1%
37	De La Salle University on 2022-10-25 Submitted works	<1%
38	Submitted on 1688509218872 Submitted works	<1%
39	University College London on 2021-07-02 Submitted works	<1%
40	University of the Philippines Diliman on 2022-06-07 Submitted works	<1%
41	df.co-aol.com Internet	<1%
42	2lo-aktywnyobywatel.pl Internet	<1%
43	Jacksonville State University on 2021-03-02 Submitted works	<1%
44	Submitted on 1688079009212 Submitted works	<1%

Sources overview

45	University of Nebraska at Kearney on 2020-02-14 Submitted works	<1%
46	Laureate Education Inc. on 2018-07-10 Submitted works	<1%
47	gpm.fkip.untirta.ac.id Internet	<1%
48	wastewise.wa.gov.au Internet	<1%
49	Jiawei Liu, Jingyuan Liu, Bin Xu, Anming Xu, Shixiang Cao, Ren Wei, Jie... Crossref	<1%
50	pinnacle.allenpress.com Internet	<1%
51	amazon.in Internet	<1%
52	publish.csiro.au Internet	<1%
53	curis.ku.dk Internet	<1%
54	studymode.com Internet	<1%
55	Laguna State Polytechnic University on 2023-01-09 Submitted works	<1%
56	University of Abertay Dundee on 2014-09-22 Submitted works	<1%

Sources overview

57	fortunebusinessinsights.com Internet	<1%
58	University of Auckland on 2023-04-05 Submitted works	<1%
59	ncbi.nlm.nih.gov Internet	<1%
60	Cebu Institute of Technology on 2022-09-16 Submitted works	<1%
61	J.M. Khatib, B.A. Herki, A. Elkordi. "Characteristics of concrete containi... Crossref	<1%
62	Universiti Malaysia Perlis on 2021-08-07 Submitted works	<1%
63	en.wikipedia.org Internet	<1%
64	vdoc.pub Internet	<1%
65	vdocuments.net Internet	<1%
66	newegg.com Internet	<1%
67	uruguaypiensa.org.uy Internet	<1%
68	Bo-Yu Peng, Yiran Li, Rui Fan, Zhibin Chen, Jiabin Chen, Anja M. Brand... Crossref	<1%

Sources overview

69	Mahidol University on 2010-05-07 Submitted works	<1%
70	Saint George's University on 2023-04-17 Submitted works	<1%
71	University of St La Salle on 2021-05-24 Submitted works	<1%
72	jcinfotr.com Internet	<1%
73	lib.dr.iastate.edu Internet	<1%
74	mafiadoc.com Internet	<1%
75	polgan.ac.id Internet	<1%
76	brimrodprimary.rochdale.sch.uk Internet	<1%
77	tdcx.com Internet	<1%
78	Technological University Of The Philippines on 2021-02-17 Submitted works	<1%
79	University of Wollongong on 2019-11-21 Submitted works	<1%
80	slideshare.net Internet	<1%

Sources overview

81	City University on 2020-10-01 Submitted works	<1%
82	Sheffield Hallam University on 2022-09-07 Submitted works	<1%
83	University of Hull on 2021-08-17 Submitted works	<1%
84	sharvielectronics.com Internet	<1%
85	A. Paulo Moreira, Jose Lima, Paulo Costa. "Improving a position contro... Crossref	<1%
86	Benowa State High School on 2020-07-24 Submitted works	<1%
87	De La Salle University - Manila on 2020-04-18 Submitted works	<1%
88	Far Eastern University on 2022-11-23 Submitted works	<1%
89	Liverpool John Moores University on 2022-03-30 Submitted works	<1%
90	University of Nebraska at Kearney on 2020-02-13 Submitted works	<1%
91	Kyu-Won Kwak, Sun Young Kim, Hyun-Jin Ko, Kyeong Yong Lee, Jeong... Crossref	<1%
92	ir.lib.uth.gr Internet	<1%

Sources overview

93	patrickbrennan.home.blog Internet	<1%
94	lazada.com.ph Internet	<1%
95	Fontana Unified School District on 2023-03-16 Submitted works	<1%
96	Soo-Jin Park, Min-Kang Seo. "Element and Processing", Elsevier BV, ... Crossref	<1%
97	University of Nebraska at Kearney on 2020-10-08 Submitted works	<1%
98	University of Witwatersrand on 2022-05-23 Submitted works	<1%
99	files.eric.ed.gov Internet	<1%
100	houzz.com Internet	<1%
101	irjmets.com Internet	<1%
102	AUT University on 2021-04-17 Submitted works	<1%
103	Atlantic Technological University on 2023-06-21 Submitted works	<1%
104	Darius Wen-Shuo Koh, Brennan Yi-Xing Ang, Joshua Yi Yeo, Zhenxiang ... Crossref posted content	<1%

Sources overview

105	Shan-Shan Yang, Meng-Qi Ding, Lei He, Chun-Hong Zhang et al. "Biode... Crossref	<1%
106	jeux-machineasous.be Internet	<1%
107	old.mb.com.ph Internet	<1%
108	portfolio.earth Internet	<1%
109	reddit.com Internet	<1%
110	Ana M. Díez-Pascual, Mohammed Naffakh, Carlos Marco, Gary Ellis. " ... Crossref	<1%
111	P.T. Harrison, R.G. Dalziel, N.A. Ditchfield, J.P. Quinn. "Neuronal-specifi... Crossref	<1%
112	Sacred Heart - Ateneo Cebu on 2019-09-24 Submitted works	<1%
113	Submitted on 1686642214237 Submitted works	<1%
114	University Tun Hussein Onn Malaysia on 2023-01-27 Submitted works	<1%
115	University of San Agustin on 2023-06-21 Submitted works	<1%
116	lucris.lub.lu.se Internet	<1%

117	repository.upnvj.ac.id Internet	<1%
118	microbiologyresearch.org Internet	<1%
119	84a69b9b8cf67b1fcf87220d0dabdda34414436b-googledrive.com Internet	<1%
120	Central Queensland University on 2017-09-14 Submitted works	<1%
121	De La Salle University on 2022-07-13 Submitted works	<1%
122	Del Siegle. "There's an App for That, and I Made It", Gifted Child Today, ... Crossref	<1%
123	EPOKA University on 2021-07-08 Submitted works	<1%
124	Melbourne Institute of Technology on 2020-06-07 Submitted works	<1%
125	Middle East College of Information Technology on 2019-06-23 Submitted works	<1%
126	Swinburne University of Technology on 2021-07-07 Submitted works	<1%
127	University of Nebraska at Kearney on 2022-02-22 Submitted works	<1%
128	hdl.handle.net Internet	<1%

Sources overview

129	mb.com.ph Internet	<1%
130	reports.valuates.com Internet	<1%
131	roborium.com Internet	<1%
132	"Web, Artificial Intelligence and Network Applications", Springer Scienc... Crossref	<1%
133	Dominik Danso, Jennifer Chow, Wolfgang R. Streit. "Plastics: Environm... Crossref	<1%
134	Edith Cowan University on 2021-05-26 Submitted works	<1%
135	L Perdanasari, B Etikasari, D L Rukmi. "Control system for temperature,... Crossref	<1%
136	Nilo M. Arago, Dariel V. De Guzman, Nicole A. De Leon, Rommel Esteve... Crossref	<1%
137	Our Lady of Fatima University on 2021-05-29 Submitted works	<1%
138	Taylor's Education Group on 2017-12-01 Submitted works	<1%
139	Universiti Malaysia Perlis on 2022-01-10 Submitted works	<1%
140	Universiti Malaysia Terengganu UMT on 2019-10-23 Submitted works	<1%

Sources overview

141	Universiti Tenaga Nasional on 2021-12-26 Submitted works	<1%
142	University of Bristol on 2021-03-17 Submitted works	<1%
143	"Emerging Research in Computing, Information, Communication and A... Crossref	<1%
144	Amin Biglari, Wei Tang. "A Review of Embedded Machine Learning Bas... Crossref	<1%
145	Anja Malawi Brandon, Shu-Hong Gao, Renmao Tian, Daliang Ning, Sha... Crossref	<1%
146	Bjorn V. Dimaya, Kevin Jerome U. Kasilag, Francis Ishmael D. Ong, Bria... Crossref	<1%
147	Cebu Institute of Technology on 2022-09-15 Submitted works	<1%
148	De La Salle University - Manila on 2020-08-17 Submitted works	<1%
149	Kingston University on 2010-09-07 Submitted works	<1%
150	North Dakota State University on 2010-07-14 Submitted works	<1%
151	Queen Mary and Westfield College on 2009-04-25 Submitted works	<1%
152	Southern Luzon State University on 2022-03-23 Submitted works	<1%

Sources overview

153	The Open University of Hong Kong on 2018-12-07 Submitted works	<1%
154	Universidad Autónoma de Nuevo León on 2018-10-14 Submitted works	<1%
155	Universiti Brunei Darussalam on 2018-04-21 Submitted works	<1%
156	Universiti Teknikal Malaysia Melaka on 2016-06-01 Submitted works	<1%
157	Universiti Teknologi MARA on 2016-10-06 Submitted works	<1%
158	Universiti Tenaga Nasional on 2019-09-21 Submitted works	<1%
159	University of Technology, Sydney on 2023-04-21 Submitted works	<1%
160	University of Westminster on 2023-05-02 Submitted works	<1%
161	Wageningen University on 2021-03-10 Submitted works	<1%
162	tup on 2022-07-18 Submitted works	<1%
163	worldciticolleges on 2023-05-02 Submitted works	<1%
164	researchsquare.com Internet	<1%

Sources overview

turnitin

Similarity Report ID: oid:25992:38621562

165	Üsküdar Üniversitesi on 2023-06-13 Submitted works	<1%

Sources overview