FARMTECH SDN. BHD.

EGG INCUBATOR
DESIGN BOOK

PRESENTED TO:

FACULTY OF MANUFACTURING AND MECHATRONICS ENGINEERING TECHNOLOGY

UNIVERSITY MALAYSIA PAHANG

PREPARED BY: GROUP 04

NO	NAME	STUDENT ID
1	WONG KAI SHENG	FB20052
2	MUHAMMAD KHALIQ IQBAL BIN MOHD RAHIPUDDIN	FA19072
3	MOHAMAD SYAHRIL SYAFIQ BIN MOHD NIZAM	FB20013
4	LEE TIAN HAU	FB20053
5	NIVYA A/P K. RAJAKUMARAN	FA20004

PREPARED FOR:
TS. DR. MUHAMMED NAFIS OSMAN ZAHID
TS. DR. NAFRIZUAN MAT YAHYA
DR. FAZILAH BINTI ABDUL AZIZ

SUBMISSION DATE: 26 JUNE 2023

EXECUTIVE SUMMARY

An Integrated Design Project (IDP) is a multidisciplinary design method that brings together the knowledge and viewpoints of many engineering disciplines to address complicated design issues. The goal of the IDP is to create a thorough design solution that fulfils a variety of goals and restrictions, as well as considering the technical, economic, and social elements of the issue.

Our IDP focuses on the development of a smart Egg Incubator which maintains optimal temperature and humidity levels, a good ventilation system, automatic incubator design and rotates automatically hence producing excellent conditions for the chick to thrive inside the fertilized egg without the presence of a hen. This project will make daily life easier for SME chicken farmers. They can use their smartphone to track the machine situation.

Technical challenges involved in the development of the egg incubator include the precise measurement of temperature, humidity and days of hatching, the effective mechanism for turning the eggs, and the fusion of the sensor and control systems into a single device. A set of sensors will be used to measure the temperature, humidity and days of hatching, and a mechanical and software combination will be used to assign the correct category after the measurements have been made. Engineers from the fields of mechanical engineering, electrical engineering, and software engineering will work together as a team to develop solutions to these problems.

The final goal of this IDP is the development of an egg incubator that can successfully hatch eggs and is dependable, efficient, and simple to use. In addition, the machine will be made with an eye toward being economically practical for the target market, which consists of farmers and businesses that process agricultural products. The device will also be made with environmentally friendly materials and manufacturing processes, with a focus on sustainability.

In conclusion, the egg incubator's IDP presents a significant challenge as well as a rare chance to combine the knowledge of various engineering disciplines to create a thorough design solution that considers the technical, financial, and social aspects of the issue.

Table of Contents

List of Figures

List of Tables

ACKNOWLEDGEMENTS

I want to express my heartfelt gratitude to the people and institutions that helped make our integrated design project a success. I would especially want to thank Dr. Nafrizuan, Dr. Fazilah, and Dr. Nafis, as well as Sir Nursyazwan and Sir Suhaimi for their efforts. Their knowledge, direction, and aid have been crucial in aiding us in achieving our goals and overcoming any obstacles that have come up throughout the project. Moreover, I want to express my gratitude to University Malaysia Pahang's Manufacturing and Mechatronic Engineering department for their aid and materials. We couldn't have achieved the high-quality outcomes we were hoping for without their dedication to creating the greatest atmosphere for our project. It took a team to complete this project, and without the help of these people and groups, it would not have been possible. We are grateful for their support because their tenacity, commitment, and enthusiasm have served as an example to us all. In closing, I want to thank everyone who has contributed to this project sincerely. We are grateful for your efforts and honored to have been a part of what you helped achieve. We will always be grateful for your contributions, and we are honored to have taken part in such a fruitful project.

CHAPTER 1

INTRODUCTION (COMPANY PROFILE)

1.1 COMPANY NAME

FARMTECH SDN. BHD.

Our company focuses on the field of agriculture in which we created our machine, the egg incubator. We strive to provide convenience, user-friendly and higher efficiency for clients to have a better way of incubating the eggs with higher possibility to hatch. Potential clients which our company targets are farmers and poultry breeders, hatcheries, educational institutions, hobbyists and small-scale breeders, research facilities and laboratories, conservation and Wildlife Organizations and entrepreneurial individuals.

To get the attention and attraction of our target clients, our company did analyses and surveys at the market. We decided to tackle a few categories of our egg incubator machine which our competitors do not have or are not very user-friendly and efficient. The categories are:

Software Programming

Our objective towards software programming of our machine is to make it as simple as possible but yet capable of taking care of a large varieties of bird eggs.

Monitoring

We also aim to provide a better monitoring system for the clients so that the users can monitor their egg incubator machine whenever they are not around with the machine which provide conveniency and higher ensuring of the eggs are being monitor well.

Setting Parameters

We made our egg incubator as simple as possible with only three buttons and one lcd for the user to have a better user-friendly experience with the machine. User can set their desired parameters like temperature, humidity, temperature tolerance, humidity tolerance, temperature and humidity thresholds, countdown days, motor hours per rotation and rotation duration for the machine to take care of the eggs with the parameters set by the users.

1.2 LOGO

Figure 1: Company Logo

1.3 ACTIVITIES

No	Date	Activity	Involvement	
1	3/4/2023	Designing of electric circuit in proteus	B + C	
2	6/4/2023	Borrow of raw materials in Machining Lab	A + B + C + D + E	
3	12/4/2023	Measuring and cutting process of the mild steel sheet, hollow mild steel, and angle bar	C + D + E	
4	12/4/2023	Borrow of electrical components	A + B + C + D + E	Wong = A
5	27/4/2023	Buying of electrical components	B + C + D	Ben = B
6	28/4/2023	Code for LCD to display introduction & slogan	B	Syahril = C
7	29/4/2023	Code for LCD to show the menu	B	Nivya = D
8	1/5/2023	Code for receving data from DHT22 Sensor & triggered Buzzer under specific conditions	B + C	Iqbal = E
9	2/5/2023	Code for conditions that triggered the bulb	B	
10	3/5/2023	Code for conditions that triggered the Humidifier	B + C	
11	4/5/2023	Code for conditions that triggered the Fan	B + C	
12	7/5/2023	Code for the system to countdown from hours, minutes and seconds	B	
13	8/5/2023	Code for conditions that triggered the Motor and rotation duration	B + C	
14	10/5/2023	Assemble and tack weld the hollow mild steel and angled bar as the frame of the machine	C + D + E	
15	10/5/2023	Code for system to follow real time clock	B	
16	11/5/2023	Code for the system to countdown from days	B	
17	12/5/2023	MIG full weld the whole frame	C + D + E	
18	13/5/2023	Code for the system to memorize the address and data	B	
19	14/5/2023	Testing simulation between arduino code and proteus	B	
20	17/5/2023	Grind the excess weld pool on the frame	C + D + E	
21	17/5/2023	Building of simple connections for each components	B + C	
22	17/5/2023	Testing of the functionality of all the components by running with new created simple coding	B + C	
23	18/5/2023	Join and weld the mild steel sheet as the base of the machine	B + C + D + E	
24	23/5/2023	Repeat of building simple connections for each components and run it with simple codes	B + C	
25	24/5/2023	Solving for one of the malfunction components and testing of components for checking confirmation	B + C	
26	24/5/2023	Connection of all the components and run it with finalize arduino code to check the whole system of our circuit.	B + C + D + E	
27	25/5/2023	Welding the mild steel sheet as the base of the machine along with the frame.	C + D + E	
28	26/5/2023	Grinding process on the welded parts	C + D + E	
29	27/5/2023	Tack welds the wire mesh and the frame together.	C + D + E	
30	27/5/2023	Drilling process on the body frame	B + C + D + E	
31	27/5/2023	Coating process for the frame of the machine	C + D + E	
32	27/5/2023	Laser cutting process for the acrylic sheet.	B + C + D + E	
33	27/5/2023	Installation of the acrylic sheet	B + C + D + E	
34	28/5/2023	Installation of the cooling fan	B + C + D + E	
35	28/5/2023	Assembly and installation of the bulb holder and bulb	B + C + D + E	
36	28/5/2023	Assembly and installation of the egg roller tray	B + C + D + E	
37	28/5/2023	Installation of the water tray and humidifier	B + C + D + E	
39	28/5/2023	Joining and managing the electrical part with the machine	B + C + D + E	
40	28/5/2023	Upload of the programming	B + C + D + E	
41	30/5/2023	Assembly of the door	B + C + D + E	
42	30/5/2023	Apply the silicone sealant between the gaps between the acrylic sheet and the frame.	B + C + D + E	
43	30/6/2023	Programming of codes for ESP8266	B	
44	30/6/2023	Designing the GUI of mobile device for the machine	B	
45	1/6/2023	Installtion of ESP8266 and wifi modem on the machine	B + C	
46	1/6/2023	Final testing after the eggs have been placed.	A + B + C + D + E	

Figure 2: Meeting of minutes

1.4 ORGANIZATION CHART

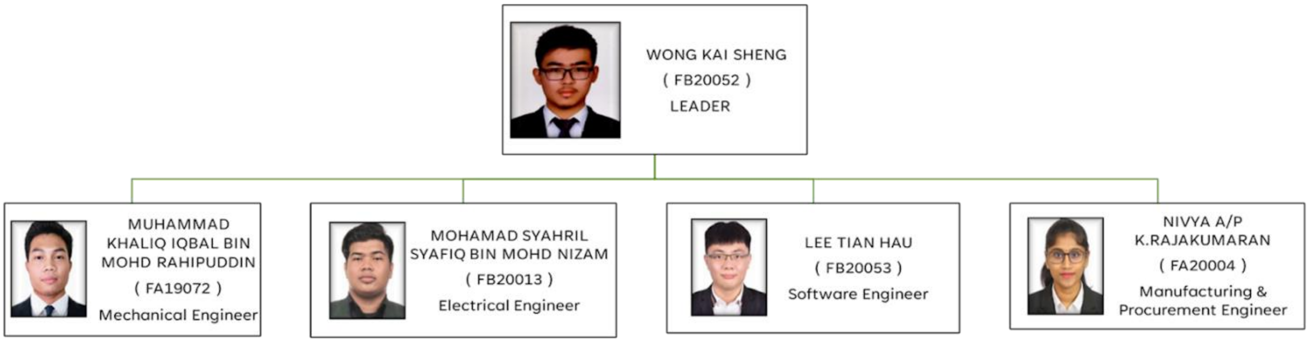


Figure 3: Organization Chart

1.5 JOB FUNCTION

Job Title	Function/Scope
Project Leader	Overall Project Management (meeting of minutes), Project Planning and Progress (Gantt chart), Project Costing and Monitoring.
Mechanical Engineer	Mechanical Analysis - FBD, Torque & Stress analysis, Thermal analysis. Mechanical Design - part design and assembly model design, detail's part drawing and assembly drawing. Computer Aided Engineering FEA Analysis (Stress and Displacement).
Electrical Engineer	Design electrical/electronic circuit - Circuit Diagram and components. Perform Circuit Analysis - Voltage, current and power analysis.
Software Engineer	Software Architecture - how the program is organized. Program Design - flow charts for microcontroller and PLC. Meaningful GUI (Graphical User Interface) or any feedback/communication tools with user.
Manufacturing & Procurement Engineer	Precedence diagram. Mass production cost analysis. Line simulation and production layout. Component specifications. Component procurement activities cost, delivery and quality. Environment and sustainability discussion.

Table 1: Job scope

CHAPTER 2

DESIGN PROBLEM STATEMENT

2.1 PROBLEM STATEMENT

Every industry in the age of globalization, including agriculture, is dependent on technology. Farmers may now run more efficiently thanks to technology. The farmers who raise chickens are having difficulty getting the eggs to hatch now. Due to the lack of signs and the inability to check the old-style incubator from the owner's house, the state of the eggs cannot be figured out. Furthermore, the yearly seasons have an impact on how a chicken develops from an embryo. The proportion of eggs hatching will decrease because of this, or the eggs will not hatch.

2.2 DESIGN SPECIFICATION & CAPABILITIES

Specification/Capabilities	**Description**
Dimension	560mm(L) x 560mm(W) x 710mm(H)
Egg Capacity	100 chicken eggs (2 Trays)
Weight	35.168 kg
Days for chicken to hatch	21 days
Features	Over/Less temperature-humidity alarm system
	Automatic On/Off Bulb according to temperature (37.7 °C)
	Automatic On/Off Humidifier according to humidity (57%RH)
	Automatic On/Off Fan according to temperature
	Automatic eggs turning according to the time
	Egg incubator data monitoring GUI
	LCD data display of temperature, humidity, motor time and days of hatching

Table 2: The specification of the machine

For other types of bird eggs

Common name	Incubation conditions			Hatcher conditions		
	Days	Temperature °F	Humidity %RH	Transfer day	Temperature °F	Humidity
canary	13–14	100.5	56–58	11	99	66–74
chicken	21	99.5	58	18	98.5	66–75
cockatiel	18–20	99.5	58–62	15–18	99	66–74
cockatoo	22–30	99.5	58–62	20–27	99	66–74
conure (sun)	28	99.5	58–62	25	99	66–74
conure (various)	21–30	99.5	58–62	18–27	99	66–74
dove	14	99.5	58	12	98.5	66–75
duck	28	99.5	58–62	25	98.5	66–75
muscovy duck	35–37	99.5	58–62	31–33	98.5	66–75
finch	14	99.5	58–62	12	99	66–74
Domestic goose	30	99.5	62	27	98.5	66–75
geese (various)	22–30	99.5	62	20–27	98.5	66–75
grouse	24–25	99.5	54–58	22	99	66–74
guinea	28	99.5	54–58	22	99	66–74
lovebird	22–25	99.5	58–62	20–22	99	66–74
macaw	26–28	99.5	58–62	23–25	99	66–74
mynah	14	100.5	56–58	12	99	66–74
parakeet	18–26	99.5	58–62	15–23	99	66–74
budgerigar	18	99.5	58–62	15	99	66–74
parrot (various)	18–28	99.5	58–62	15–25	99	66–74
parrot (african grey)	28	99.5	58–62	25	99	66–74
chukar partridge	23–24	99.5	62	20	99	66–74
peafowl	28–29	99.5	58–62	25–26	98.5	66–75
ptarmigan	21–23	99.5	58–62	18–20	99	66–74
raven	20–21	99.5	58–62	17–18	99	66–74
ring-neck pheasant	24–24	99.5	58–62	21	99	66–74
pheasant	22–28	99.5	58–62	20–25	99	66–74
pigeon	17–19	100.5	58	14	99	66–74
bobwhite quail	23	99.5	54–58	21	99	66–74
japanese quail	17–18	99.5	58–62	15	99	66–74
swan	33–37	99.5	58–62	30–33	99	66–74
turkey	28	99.5	54–58	25	98.5	66–75
emu	49–50	97.5	32–40	47	97.5	69
ostrich	42	97.5	32–40	39	97.5	69
rhea	36–42	97.5	50	34–37	97.5	69

Figure 4: The temperature and humidity level for different types of eggs

According to the table figure above, the requirements criteria for the eggs to hatch as stated above are achievable by our machines as the temperature and humidity value can be reached and maintained by our egg incubator after we have tested the range of the temperature and humidity the egg incubator can maintain.

CHAPTER 3
MECHANICAL DESIGN

3.1 INTRODUCTION

Mechanical design refers to the process of creating various mechanically related elements, such as parts, pieces, systems, or products. When designing mechanical systems, it is essential to consider key requirements such as functionality, safety, dependability, manufacturability, weight, size, wear, maintenance, and liability. Also, there are many other criteria that may be considered. While the specific standards and criteria may vary from one product design to another, certain fundamental design concerns remain applicable to all mechanical systems. These concerns include load capacity, deformation, stability, and durability. To optimize products, it is crucial to model and analyze how these evaluation criteria depend on design variables. Finite element analysis (FEA) is a valuable tool for examining mechanical systems and evaluating their loading capability, stability, and fatigue life. Furthermore, this chapter also delves into the governing mathematical models used in typical engineering design problems.

3.2 MECHANICAL ANALYSIS
3.2.1 MATERIAL SELECTION

When it comes to designing and fabricating a product, the first and foremost important factor to consider is material selection. Before proceeding with the selection process, it is crucial to identify the type of material used in creating the desired product. In our case, we have considered two different options: aluminum and stainless steel. To determine the most suitable material, a thorough assessment and evaluation of their compatibility with the manufacturing process is necessary. The following table provides a comparison between aluminum and stainless steel. As manufacturers of egg incubators, we prioritize functioning as a key factor in our decision-making process.

3.2.1.1 FRAME AND BODY MATERIAL

Initially, our plan was to utilize a stainless-steel funnel for its grade 316 properties which makes it much more corrosion resistant, which makes it suitable for manufacturing egg incubators. Moreover, it offers easy accessibility. However, considering the high cost involved, we made the decision to use for a Rectangular Hollow Bar instead. The table below provides an overview of the frame materials we selected for this project:

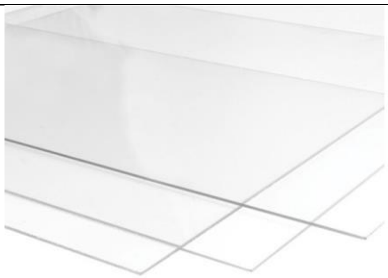
	Material	Justification
1	 Hollow Steel	For the frame we use rectangular hollow mild steel material. This material is quite good and strong, also the best choice for the agriculture industry because of its good heat resistance. This material is also cheap and don’t need to buy as it can be gotten in the FTKPM Lab
2	 Sheet Metal Plate	For the base part, we chose sheet metal plate used to support the egg incubator's upper body. This material is quite good and strong, also the best choice for agriculture because it is heat resistant. This material is also cheap and don't need to buy it as it can be bought in the FTKPM Lab.
3	 Acrylic Perspex Sheet	For the body part, the second material that we choose is acrylic Perspex sheet. It is also used to cover the body of the egg incubator. This material is transparent but tough, also the best choice for agriculture because of its transparency. This material is also light compared to the mild steel and you don't need to buy it as it can be bought in the FTKPM Lab.

Table 3: Type of material used for the Egg Incubator

Aluminum	Mild steel
More expensive	Less expensive
Corrode but not rust	Easily rust
Lower density	Higher density
More malleable and elastic.	Less malleable and elastic.
Light weight	Heavy
Tensile strength is 276 MPa	Tensile strength is 400 MPa
Thermal conductivity at 125°C is 255W/m°C	Thermal conductivity at 125°C is 51W/m°C

Table 4: Comparison between Aluminum and Mild Steel

3.1.1.2 REFLECTION

We can see that there is a difference between the two materials based on the table above. Although mild steel is less expensive than aluminum, aluminum is more malleable and elastic than mild steel. Aluminum can be easily formed but is difficult to join with welding, according to my observations. Only aluminum can be welded using certain welding procedures. As a result, mild steel is a simple material to use for MIG welding. In comparison to mild steel, cutting the material takes less time. Because our product is frequently exposed to sunshine, it has a higher potential to corrode, so aluminum is the superior material.

On the other hand, the product can be made of stainless steel, but it will require an additional manufacturing procedure, such as coating to avoid corrosion, such as powder coating in surface finishing. Stainless steel has a tensile strength of 400 MPa compared to 276 MPa for aluminum. This demonstrates that stainless steel has greater strength than aluminum. Because our product only needs to monitor the soil's humidity, it does not require any additional strength to operate. In terms of thermal conductivity, a lower number is required to prevent the body part from becoming overheated at 125°C cooking temperature. In conclusion, stainless steel outperforms aluminum in terms of strength, cost, and heat conductivity.

3.2.2 STRUCTURE DESIGN

3.2.2.1 DETAIL DRAWING WITH COMPLETE DIMENSION

The study and product testing carried out prior to the product's fabrication informed us of the component choices we made for the project. The specifics of the component selection drawing, as displayed in the Appendices.

3.2.2.2 3-DIMENSIONAL DRAWING OF STRUCTURE DESIGN

The final design concept is the stage at which we have completed our conceptual design project selection. CATIA software was used to undertake this final design idea for further observation and analysis before fabricating our product, the Egg Incubator. The picture below depicts our product's final decision.

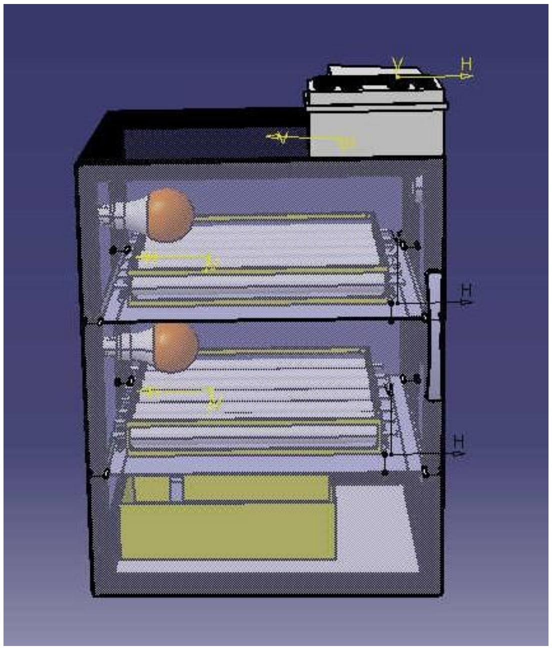


Figure 5: Front view of the machine

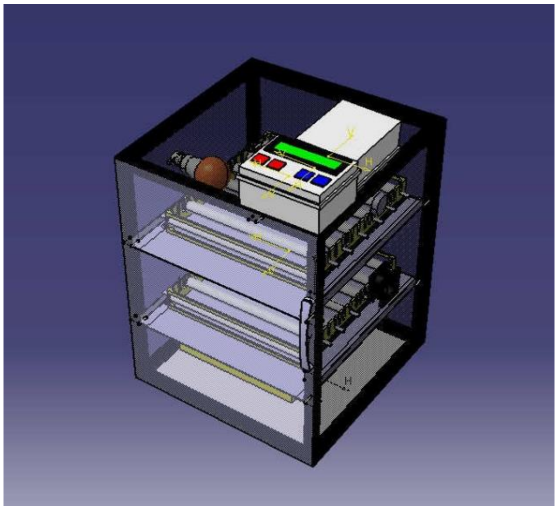


Figure 6: Isometric view of the machine

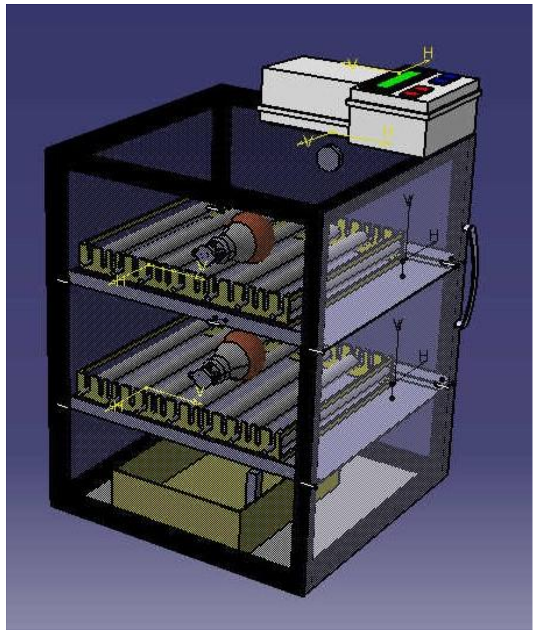


Figure 7: Isometric view from the left side of the machine

3.2.3 MECHANISM DESIGN

The Egg incubator incorporates a variety of mechanisms, each serving a distinct purpose to ensure its smooth operation.

3.2.3.1 EGG ROTATION TRAY

The figures below show the assembly of the egg rotation tray. The egg will be placed in between on the top of the roller cylinder. The supply is controlled by servo motors to rotate the eggs.

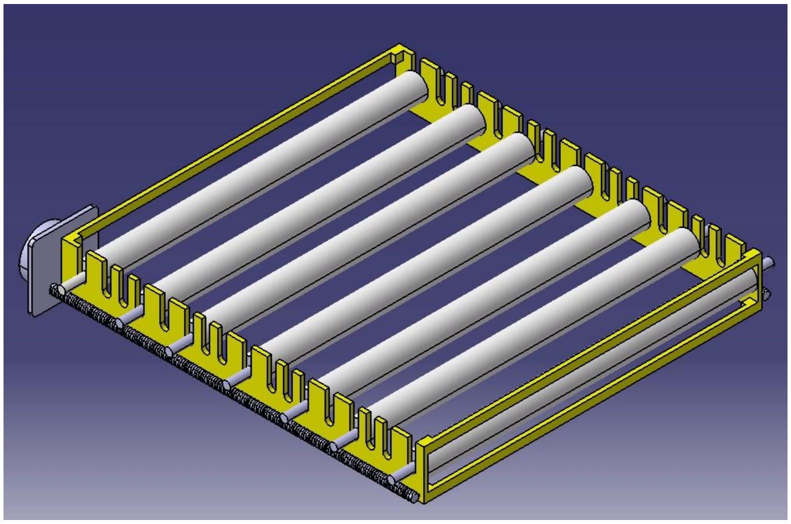


Figure 8: The view of egg rotation tray

3.2.3.2 VENTILATION FAN

The figures below show the ventilation fan. It is attached on one side of the acrylic sheet to be installed on the Egg incubator's frame body. It is used to regulate the temperature inside the Egg incubator and to reduce humidity, controlled by main switch.

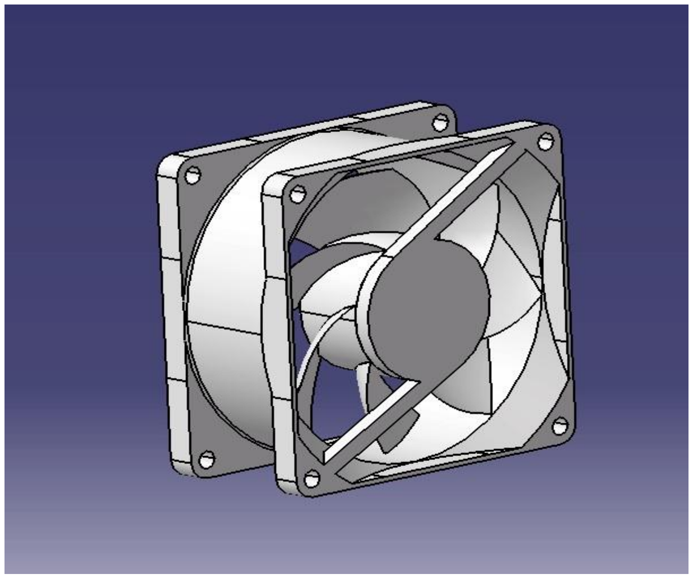


Figure 9: The view of ventilation fan

3.2.4 MECHANICAL CALCULATION

Power motor analysis

<u>Power of motor</u>

$P = VI$

$P = (12V)\,(0.3A)$

$P = 3.6\ W$

<u>Torque of motor</u>

Torque=Power/speed

T = P/W

T = 3.6 W / 5 RPM

T = 3.6 W / 0.5236 rad/s

T = 6.88 Nm

Torque required to rotate the gears and shafts

Total mass of the eggs= 58.665g x 50 = 2933.25g

= 2.9333kg

Radius of gears = 0.002m

Radius of shaft = 0.002m

Moment of inertia at the gear,

I = 1/2m(r2gear=r2shaft)

I = ½ (2.9333) [(0.002^2) +(0.002^2)]

=0.00001173kgm^2

RPM of the gear = 5 rpm, gear ratio = 1

Angular velocity of the gear =

= 0.5236 rad/s

Time to reach 0 rpm to 5 rpm in 12 seconds.
Angular acceleration of the gear, α = (0.5236 - 0) / 12

= 0.04363 rad/s^2

Torque required to rotate gears and shafts, $\boldsymbol{\tau}$
= Total Moment of Inertia x Angular acceleration

= 0.00001173 x 0.04353

$= 5.1061x10^{-7}$Nm

Power of motor

P = VI

P = (12V) (0.3A)

P = 3.6 W

Torque of motor

Torque=Power/speed

T = P/W

T = 3.6 W / 5 RPM

T = 3.6 W / 0.5236 rad/s

T = 6.88 Nm

Torque required to rotate the gears and shafts

Total mass of the eggs= 42.5g x 17= 722.5g

=0.7225kg

Radius of gears = 0.002m

Radius of shaft = 0.002m

Moment of inertia at the gear,

I = 1/2m(r2gear=r2shaft)

I = ½ (0.7225) [(0.002^2) +(0.002^2)]

$=0.00000289kgm^2$

RPM of the gear = 5 rpm, gear ratio = 1

Angular velocity of the gear =

= 0.5236 rad/s

Time to reach 0 rpm to 5 rpm in 12 seconds.

Angular acceleration of the gear, α = (0.5236 - 0) / 12

= 0.04363 rad/s^2

Torque required to rotate gears and shafts, $\boldsymbol{\tau}$

= Total Moment of Inertia x Angular acceleration

= 0.00000289 x 0.04353

= 1.2580x10 $^{-7}$Nm

Light bulb analysis

Volume inside incubator

0.5 x 0.5 x 0.7 = 0.175 m3

Mass of air inside incubator

P = M/V, where p is note as density of air at 37.5 0c = 1.137 kg/m3

M denote the mass of air (kg), V = volume of air inside

M = P x V

M = 1.137 x 0.175

M = 0.198975 kg

Heat energy amount calculation

for tray Q = MC∆T

M = 0.921 kg, C = 1.67 kJ/kg

Q = 2 x 0.921 x 1.67 x (37.5-26)

$Q = 35.38$ kJ

Heat energy amount calculation

For air inside container $Q = MC\Delta T$

C of air at 37.5 °C = 1.007 kJ/kg 0c

$Q = (0.198975)\ (1.007)\ (37.5\text{-}26)$

$Q = 2.3042$ kJ

Heat energy amount calculation

For egg $Q = MC\Delta T$

Mass of egg = 0.06 kg

C = 3.182 kJ/Kg 0c

$Q = (0.06 \times 84)\ (3.182)\ (37.5\text{-}26)$

$Q = 245.90$ kJ

Heat energy amount calculation

For water (1 litre) $Q = MC\Delta T$

C = 4.187 kJ/Kg 0c

$Q = (1)\ (4.187)\ (37.5\text{-}26)$

$Q = 48.1505$ kJ

Heat energy amount calculation

For steel body $Q = MC\Delta T$

C = 0.420 kJ/Kg 0c

$Q = (11.696)\ (0.42)\ (37.5\text{-}26)$

Q = 56.49186 kJ

Total amount of heat energy

Q total = 35.38 + 3.3042 + 2544.90 + 48.1505 + 56.49168

Q total = 388.22 kJ

Electric power required to be

P = Q/t

P = 388.2247/(24x60x60)

P = 0.004493 kW @ 44.493 W

The total power needed to generate heat inside incubator for reaching temperate 37.5-degree Celsius is about 45-watt power

3.3 FINITE ELEMENT ANALYSIS

Finite Element Analysis, or FEA, is a mathematical tool for analyzing and measuring the effects of real-world conditions on a component or assembly. Engineers can utilize these simulations, which are done using specialized software, to uncover design defects such as strain zones and weak places. We use FEA software to reduce the number of physical prototypes and testing and optimize components throughout the design phase, allowing us to manufacture better products faster and for less money. For the FEA, we used CATIA software.

The act of modeling a part or assembly's behavior under specified conditions to examine it using the finite element method is known as finite element analysis (FEA) (FEM). Engineers use FEA to model physical phenomena and eliminate the need for physical prototypes while allowing for component optimization as part of the project design process.

Our project employs FEA because it can assist us in determining the minimum and maximum load or force that can be supported. The Von Mises stress is a measurement that can be used to evaluate if a material will yield or fracture. It's commonly used on ductile materials like metals. The von Mises yield criterion asserts that a material will yield if its von Mises stress under load is equal to or greater than the yield limit of the same material under simple tension.

3.3.1 ASSUMPTION OF CRITICAL PARTS

We have assumed 3 parts as our product critical part. The parts are frame body, the L-shaped angular steel holder for egg tray, and the mild steel sheet for floor or base part of the Egg Incubator.

3.3.1.1 FRAME BODY

For the frame body, we have put a maximum load of 4.5kg or 45N of load because we assume that the maximum load of acyclic sheet and junction box that place on top of body is approximately near to 4.5kg. The table below shows the details of the Finite Element Analysis of the frame body:

Type of stress	**Minimum**	**Maximum**
Von mises stress	198 N/m^2	1.98e + 003 N/m^2
Displacement	2.92e-007 mm	2.92e-007 mm
Yield Strength		2.5e+008 Nm2

Table 5: Load Analysis by the type of stress

The figures below show the simulation made in CATIA to do Finite Element Analysis.

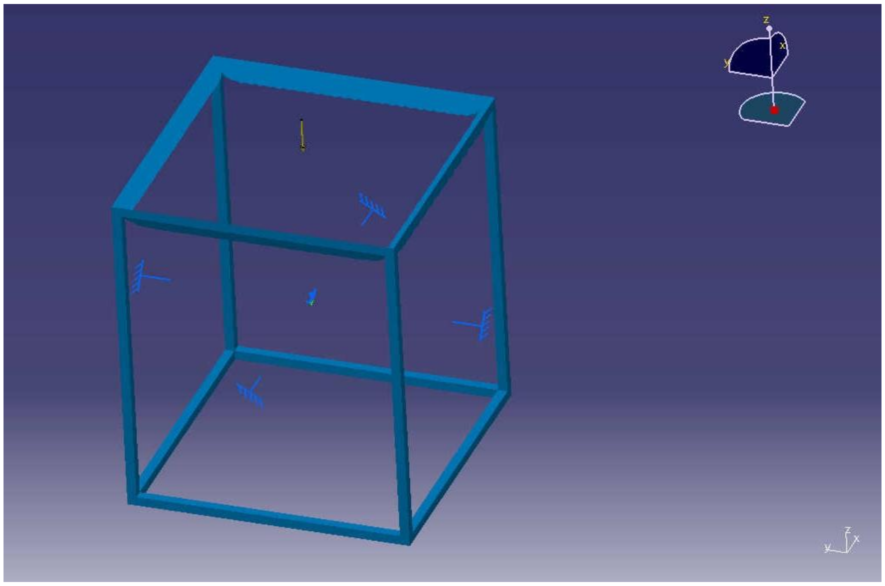


Figure 10: The clamps and load exerted on the frame body

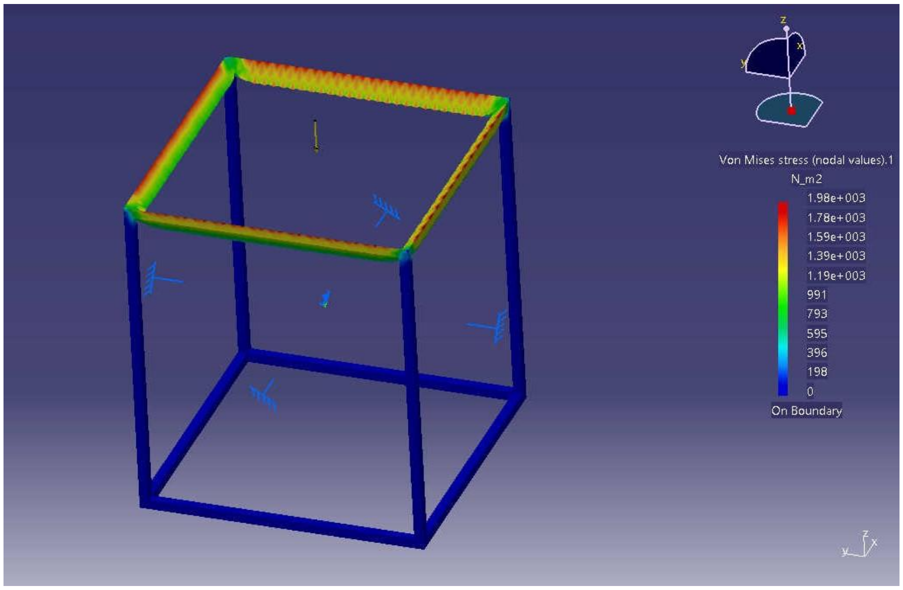


Figure 11: Von Mises stress results of the frame body

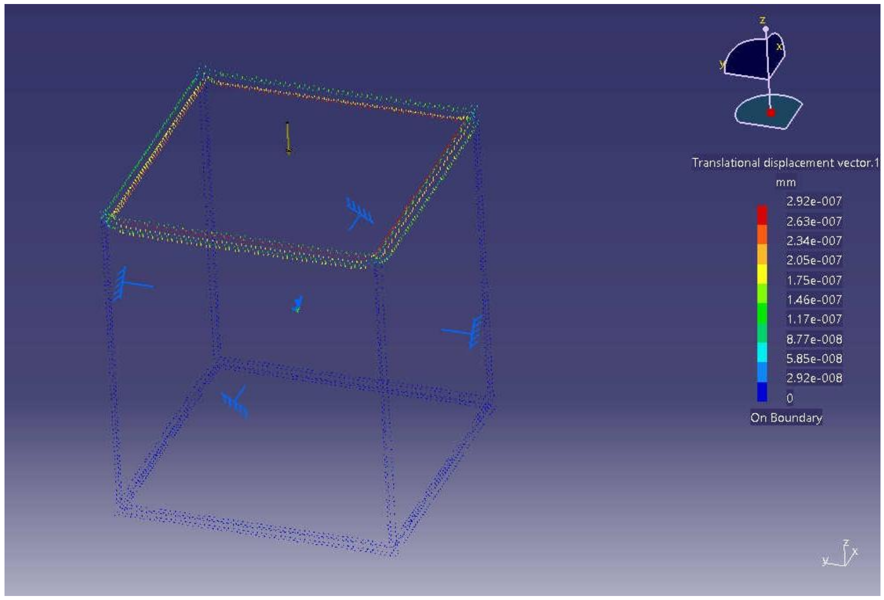


Figure 12: Displacement stress result of the frame body

3.3.1.2 BASE PART

For the base part, we have put a maximum load of 30kg or 300N of load because we assume that the maximum load that place on top of lower base part (mild steel) is approximately near to 30kg. The table below shows the details of the Finite Element Analysis of the frame body:

Type of stress	**Minimum**	**Maximum**
Von mises stress	195e + 005 N/m^2	1.95e + 006 N/m^2
Displacement	0.00164 mm	0.0164 mm
Yield Strength		2.5e+008 Nm2

Table 6: Load Analysis by the type of stress

The figures below show the simulation made in CATIA to do Finite Element Analysis.

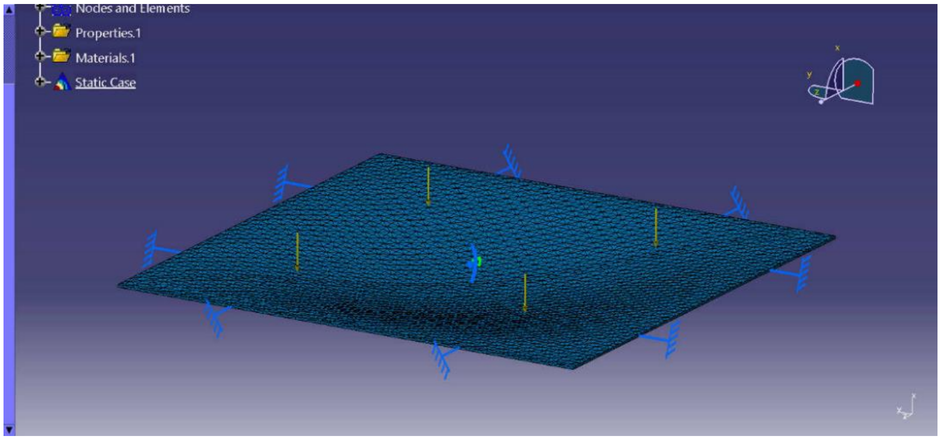


Figure 13: The clamps and load exerted on the lower base part

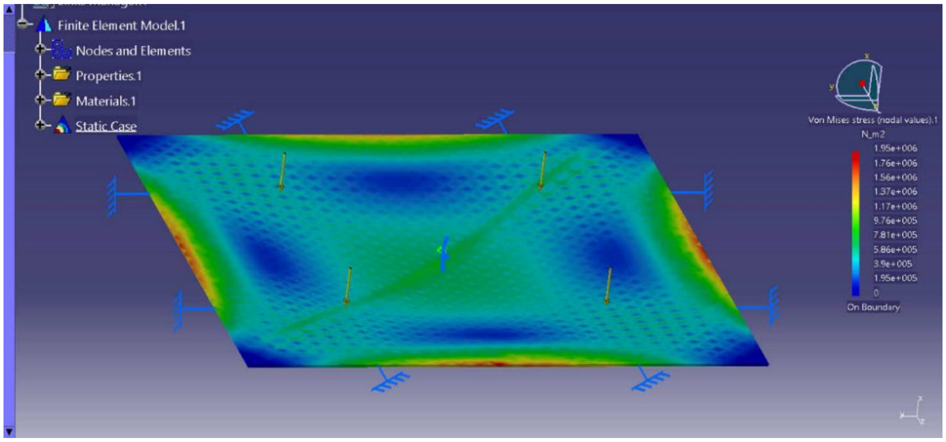


Figure 14: Von Mises stress results of the lower base part

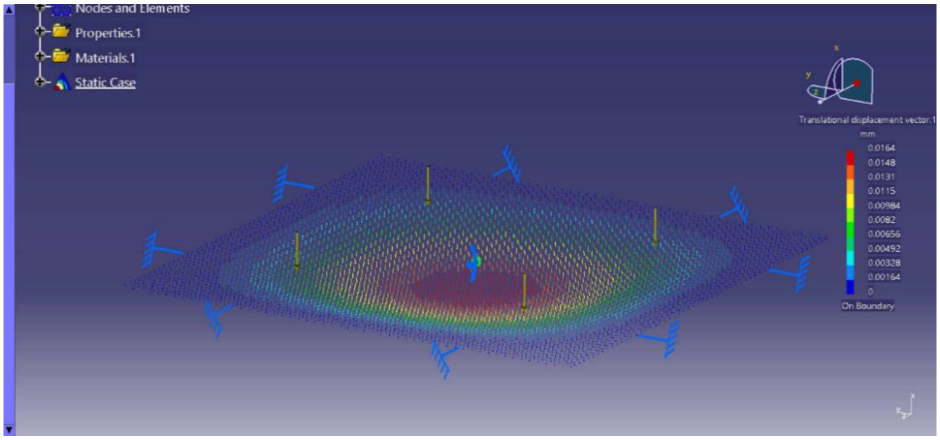


Figure 15: Displacement stress result of the lower base part

3.3.1.3 L SHAPE ANGULAR STEEL

For the L shape angular steel, we have put a maximum load of 4.5kg or 45N of load because we assume that the maximum load that places on L shape angular steel is approximately near to 4.5 kg. The table below shows the details of the Finite Element Analysis of the frame body:

Type of stress	**Minimum**	**Maximum**
Von mises stress	195e + 004 N/m^2	1.59e + 005 N/m^2
Displacement	5.32e-006 mm	5.32e-005 mm
Yield Strength		2.5e+008 Nm2

Table 7: Load Analysis by the type of stress

The figures below show the simulation made in CATIA to do Finite Element Analysis.

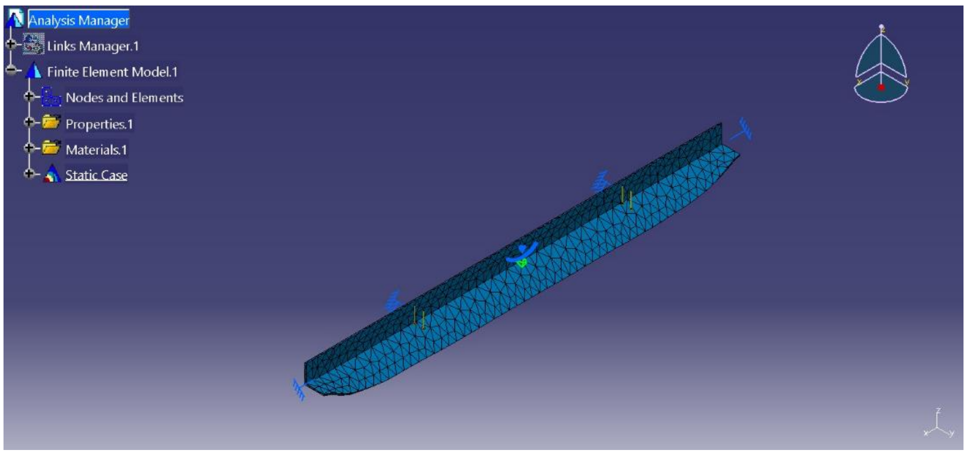


Figure 16: The clamps and load exerted on the L shape angular steel.

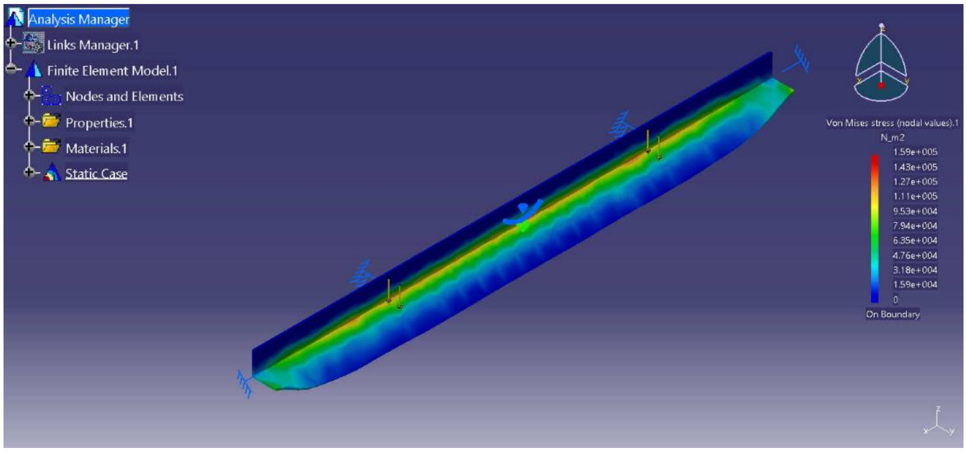


Figure 17: Von Mises stress results of the L shape angular steel

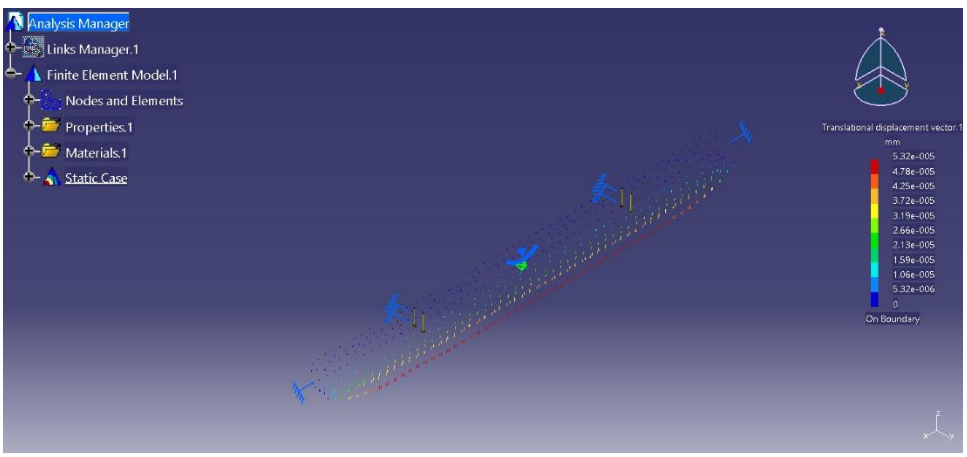


Figure 18: Displacement stress result of the L shape angular steel

CHAPTER 4

ELECTRICAL/ELECTRONICS DESIGN

4.1 INTRODUCTION

The project's electrical component entails identifying and connecting all electrical parts required for the project prototype. The parts must be seamlessly connected with the project's mechanical and software components and are chosen depending on the machine's intended functions. The electrical design must take power efficiency and suitable connections into account to ensure a well-functioning circuit and prevent any faults or component damage.

Before deciding on the power supply, the type of motor, sensors, and microcontroller must be chosen to make sure that the power input is sufficient to support the power requirements of all components.

4.2 OBJECTIVES

I. Design an electric circuit that fits and fulfils the requirements of the project.

- To guarantee that the components function as needed, the electrical design must take into account the machine's desired functioning. Before the circuit is built, the specs can be checked using simulation software like Proteus or Multisim.

II. Use and choose more relevant electrical components.

- Considerations like cost, power usage, and accessibility should be made when choosing the components. Industry standards, such as those produced by the Institute of Electrical and Electronics Engineers (IEEE), may serve as guidelines for the selection process.

III. Design a circuit that can power up all the components in the circuit.

- The power supply needs to be strong enough to support the circuit's components' power output. Before making a choice, it's critical to have a solid understanding of power consumption and the various power supply options. To determine the amount of power needed for each component in the circuit, a power consumption analysis should be carried out.

IV. Do the full power consumption analysis of the machine.

● Software like Power analyzer, which enables users to model and analyze the power consumption of the complete machine, can be used to undertake a power consumption analysis. This study is essential to make sure that the power supply option is capable of supporting the machine's power requirements and avoiding component failures because of power supply constraints. To make wise choices regarding the power supply and its components, the analysis' findings should be employed.

4.3 SCHEMATIC DIAGRAM

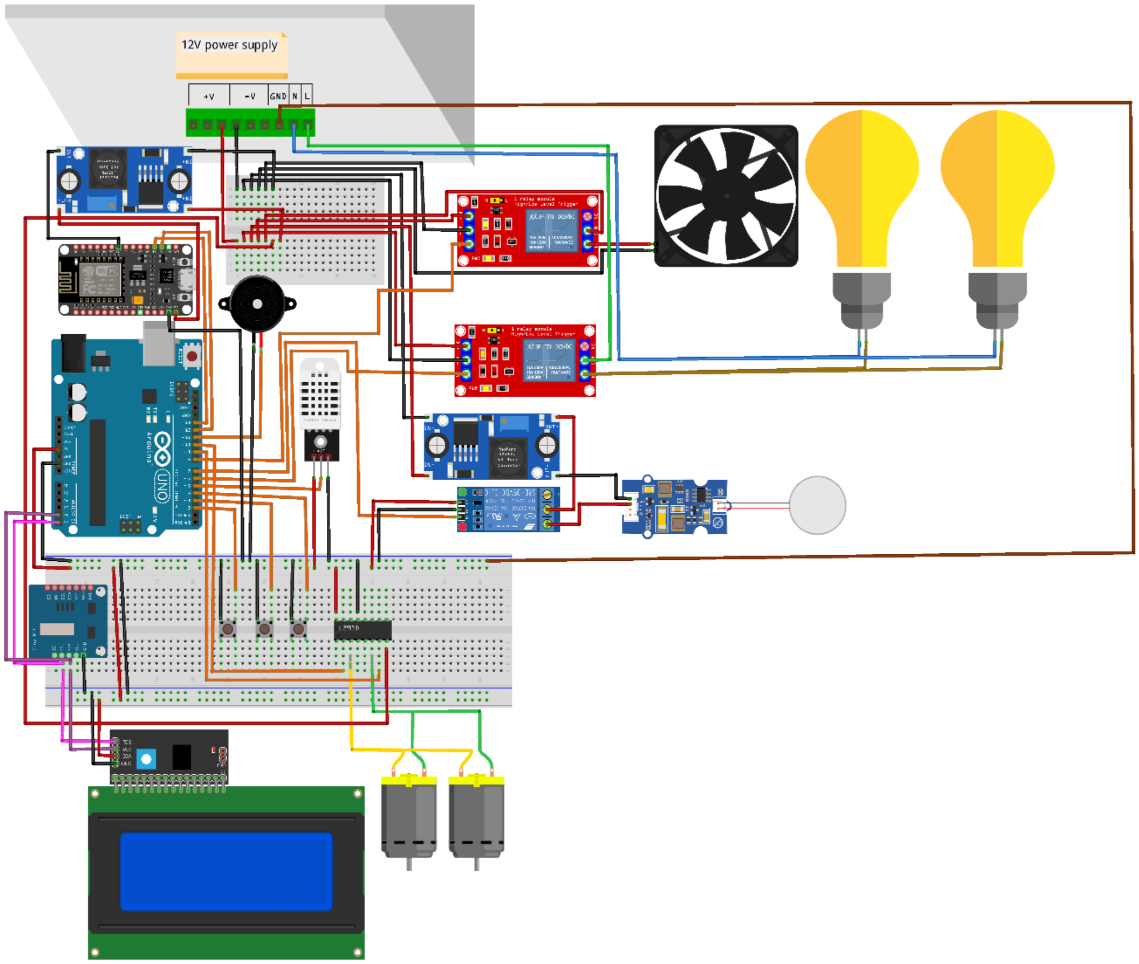


Figure 19: Schematic Diagram of Egg Incubator

4.4 LIST OF ELECTRICAL COMPONENTS

No	Components	Quantity
1	Arduino Uno R3	1
2	DHT22	1
3	Buzzer	1
4	LCD Display 20x4	1
5	Cooling Fan	1
6	Motor Driver L293D	1
7	Humidifier	1
8	Light Bulb	2
9	Push Button	3
10	Synchronous Motor	2
11	Power Supply	1

Table 8: List of Electrical Components

No	Components	Specification
1	Arduino Uno R3 	Circuit Operating Voltage: 5V Input Voltage: 7V - 12V
2	DHT22 	Voltage: 3.5V - 5.5V Current: 2.5mA Temperature Range: -40’C - 125’C Humidity Range: 0% - 100%

3	Buzzer 	Voltage: 3V – 30V Current: 0.054A
4	LCD Display 20x4 	Voltage: 5V Current: 50mA
5	Cooling Fan 	Operating Voltage: 12V Current: 0.30A
6	Humidifier 	Voltage: 3V – 12V Current: **300m**A
7	Light Bulb 	Voltage: 50V Power: 75W
8	Push Button 	Shaft/Mounting hole: 12mm(diameter) x 7.5mm (length) Dimension: **21.6mm x 17.5mm** x 17.5mm (include terminal)

9	Synchronous Motor 	Voltage: 12V Speed: 5RPM Current: 0.3A
10	Power Supply 	Voltage: 12V Current: 10A

Table 9: List of Electrical Components with Specifications

4.5 POWER CONSUMPTION ANALYSIS

It relates to the mathematical evaluation of electrical properties. We can assess whether the generated power is sufficient to power all components and whether the input voltage will not damage the electronic components through this evaluation. We may also calculate the project's energy consumption. The procedures for doing electrical analysis are presented.

4.5.1 POWER CONSUMPTION CALCULATION

The estimation calculation of power that will be used, including losses, in this electrical calculation section. This is to demonstrate that our circuit will not be overloaded and will function properly.

I. Arduino Uno R3

Voltage: 5V

Current: 0.38A

Power = (5V × 0.38A)

= 1.9W

II. DHT22

Voltage: 5.5V

Current: 2.5mA

Power = (5.5V × 2.5mA)

= 0.0138W

III. Buzzer

Voltage: 5V

Current: 0.054A

Power = (5V × 0.054 A)

= 0.27W

IV. LCD Display 20x4

Voltage: 5V

Current: 50mA

Power = (5V × 50mA)

= 0.25W

V. Cooling Fan

Voltage: 12V

Current: 0.3A

Power = (12V × 0.3A)

= 3.6W

VI. Humidifier

Voltage: 12V

Current: 0.3A

Power = (12V × 0.3A)

= 3.6W

VII. Light Bulb

Voltage: 50V

Current: 1.5A

Power = (50V × 1.5A)

= 75W

VIII. Synchronous Motor DC 12V

Voltage: 12V

Current: 0.6A

Power = (5V × 0.3A)

= 3.6W

No	Components	Quantity	Voltage(V)	Current(A)	Power(W)
1	Arduino Uno R3	1	5	0.38	1.9
2	DHT22	1	5.5	2.5m	0.0138
3	Buzzer	1	5	0.054	0.27
4	LCD Display 20x4	1	5	50m	0.25
5	Cooling Fan	1	12	0.3	3.6
6	Humidifier	1	12	0.3	3.6
7	Light Bulb	2	50	1.5	75
8	Synchronous Motor DC 12V	2	12	0.3	3.6
P_{out}					91.83 W

Table 10: Power Consumption Calculation Details

Power supplied from Power Supply,

$P_{in} = (V_{in})(I_{in})$ = (12V) (10A) = 120W

Total power, P_{in} = 120 W

As a result, $\boldsymbol{P}_{in}$>$\boldsymbol{P}_{out}$, therefore there are no overload and fall in the safe zone.

Power Loss, $\boldsymbol{P}_{loss}$= $\boldsymbol{P}_{in}$ - $\boldsymbol{P}_{out}$,

= 120 W – 91.83 W

= 28.17 W

Energy Efficiency, η = (P_{out}/P_{in} X 100) %

η = (91.83/120 X 100)

η = 76.52%

4.6 ELECTRICITY COST CALCULATION

The calculation is based on the industrial tariffs and assumes the machine will run for 1 month.

Non-Residential Customers	ICPT surcharge rate (RM/kWh)
Micro, Small and Medium Enterprises (MSMEs) (Tariff B, D, H, H1 and H2) Low Voltage Commercial | Low Voltage Industrial | Low & Medium Voltage Specific Agriculture	NO CHANGES 0.0037
Non-MSMEs Streetlight | Mining | Medium & High Voltage Commercial | Industrial	0.20

Figure 20: Industrial Tariffs

Using RM0.20 /kWh for the calculation.

Total Operating Hours per Month = 24 hours × 7 days × 4 weeks = 672 hours

Total Power Consumption per Month = Total Operating Hours per Month × Total Power Supplied

= 672 hours × 0.076525 kWh

= 51.4248 kWh

Total Cost of Power Consumption per Month

= 51.4248 kWh × RM 0.20

= **RM 10.28**

4.7 CONCLUSION

As a conclusion, the circuits can supply enough power to all electrical components. The power consumption of the whole machine is reasonable, and less power consumption compared to our daily life electrical appliances which are egg incubators.

CHAPTER 5
SOFTWARE DEVELOPMENT

5.1 INTRODUCTION

Egg incubator software development using Arduino IDE combines the power of Arduino boards with the simplicity of Arduino programming language to create intelligent and customizable egg incubators. By writing code in Arduino IDE, developers can control various aspects of the incubator, such as temperature, humidity, turning mechanisms, and monitoring systems. Arduino IDE provides a user-friendly interface, libraries, and examples, making it accessible for beginners and experienced programmers alike. With Arduino as the brain of the incubator, the software can process sensor data, make decisions based on programmed logic, and control the necessary components to provide optimal conditions for successful egg hatching.

5.2 SOFTWARE ARCHITECTURE (BLOCK DIAGRAM)

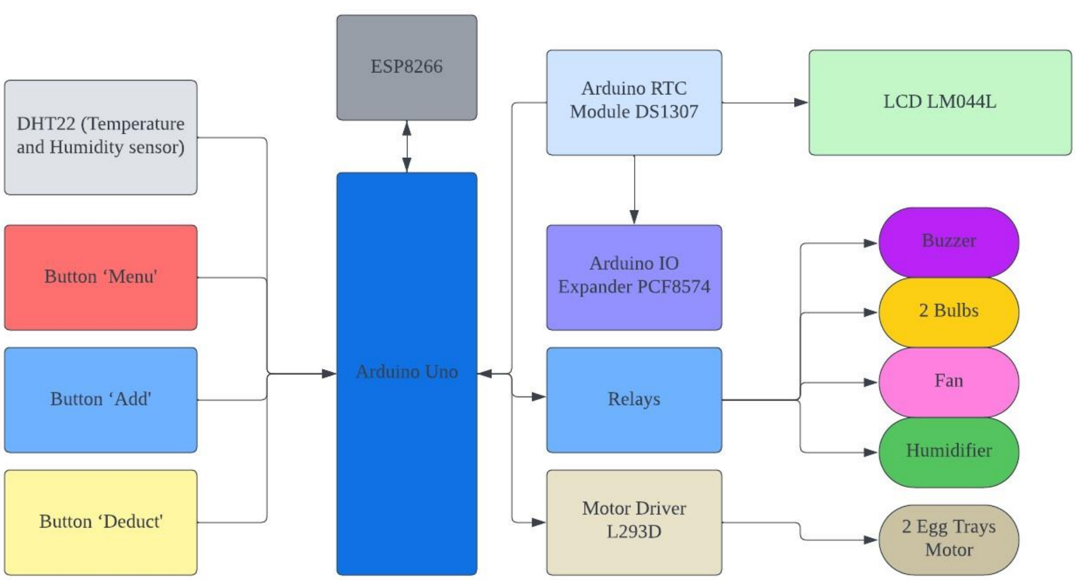


Figure 21: The flowchart of the software architecture

5.3 PROGRAMMING FLOW CHART

According to our group's objectives, we applied it to the egg incubator. We have separated the flow charts into a few categories for the client to have a better understanding of our machine's work. The few categories of flow charts are operation process, parameters setting and graphical user interfaces flowcharts.

5.4 EGG INCUBATOR OPERATION PROCESS FLOWCHART

The flow chart below shows how our egg incubator takes care of the eggs according to the sensor (DHT22), countdown timer and countdown days. Each time the DHT22 sends value, the machine processes and reads the value whether it is within the range of the desired value that we have set. Then the machine will operate each component based on the condition of the value range. Meaning that, it will turn on light bulbs if temperature value from DHT22 is lower than tolerance value and turns on the fan if the temperature is higher than the tolerance value. Same goes to the humidifier, if the humidity is lower than the tolerance value, the humidifier will turn on, but it does nothing when humidity is higher than tolerance value as we found out that humidity drops rapidly so we do not program the fan to be turned on. The buzzer will be turned on if the temperature and humidity is not within the threshold value. The egg trays motor will start to turn on to rotate the eggs if the countdown of the hours is zero and the lcd will display the eggs has been hatched when the countdown days has reached zero same goes with buzzer will be turn on also. The machine will keep on continue to take care of the eggs even if the countdown days have reached to ensure the eggs are still not hatching and will still be taken care of.

Figure 22: The egg incubator operation flowchart

5.5 EGG INCUBATOR PARAMETERS SETTING FLOWCHART

The flow chart below shows the process of how we can set the parameters of temperature reference value, temperature tolerance value, humidity reference value, humidity tolerance value, temperature threshold value, humidity threshold value, countdown days, hours for each rotation and rotation duration based on what types of eggs we want to incubate.

Each time we pressed "Next" button of the machine, the lcd will bring us to each value that we can adjust to set the desired value suitable for taking care of the eggs that used.

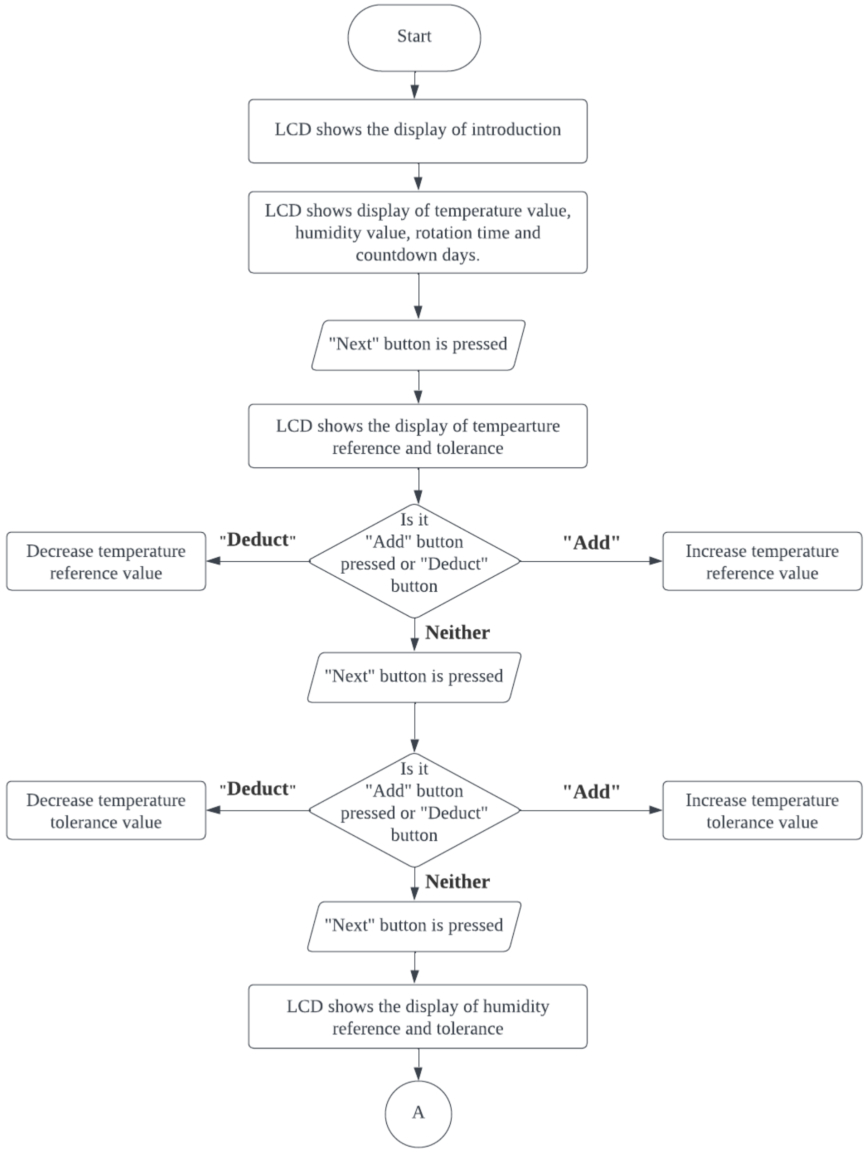


Figure 23: Egg incubator parameters setting flowchart

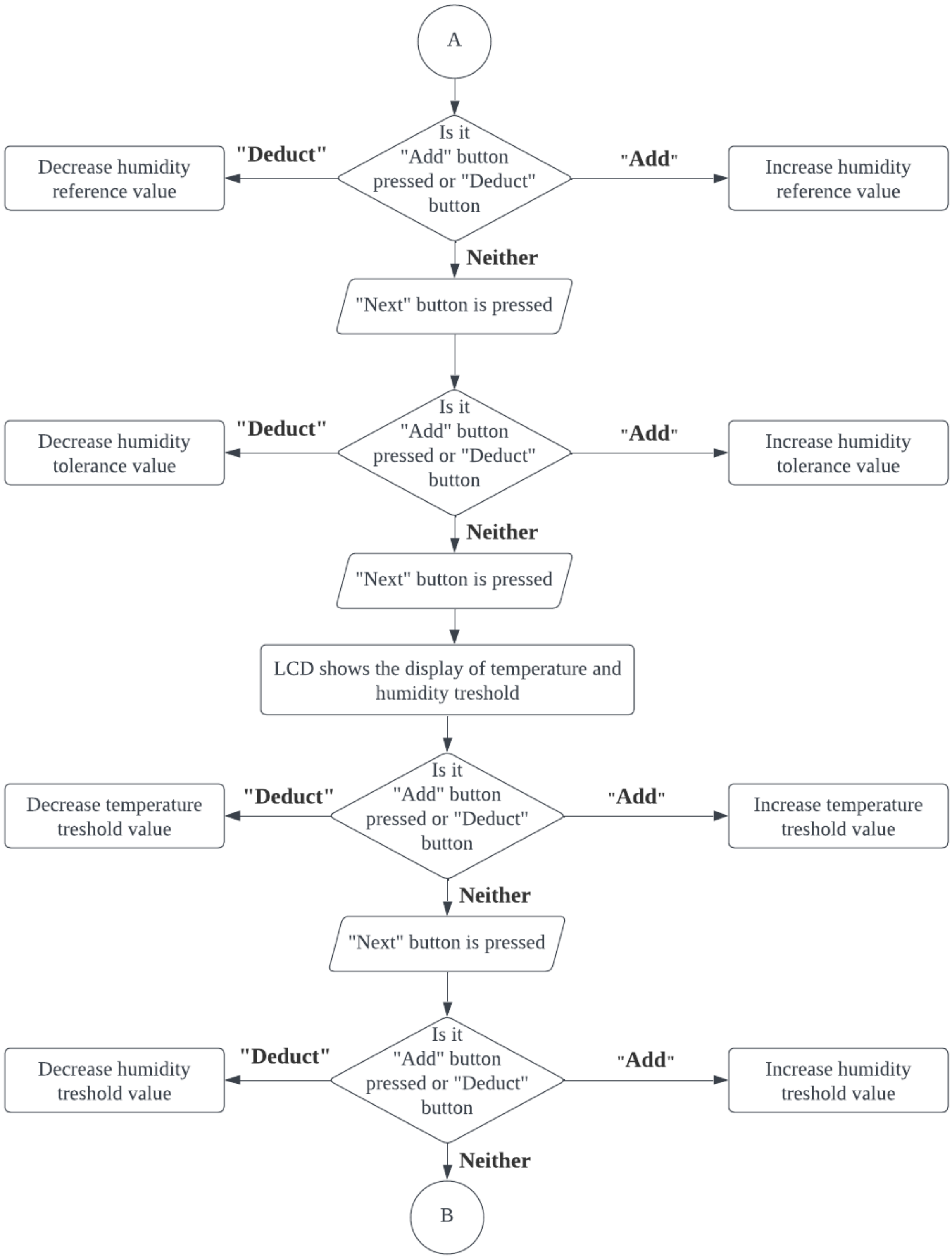


Figure 24: Egg incubator parameters setting flowchart

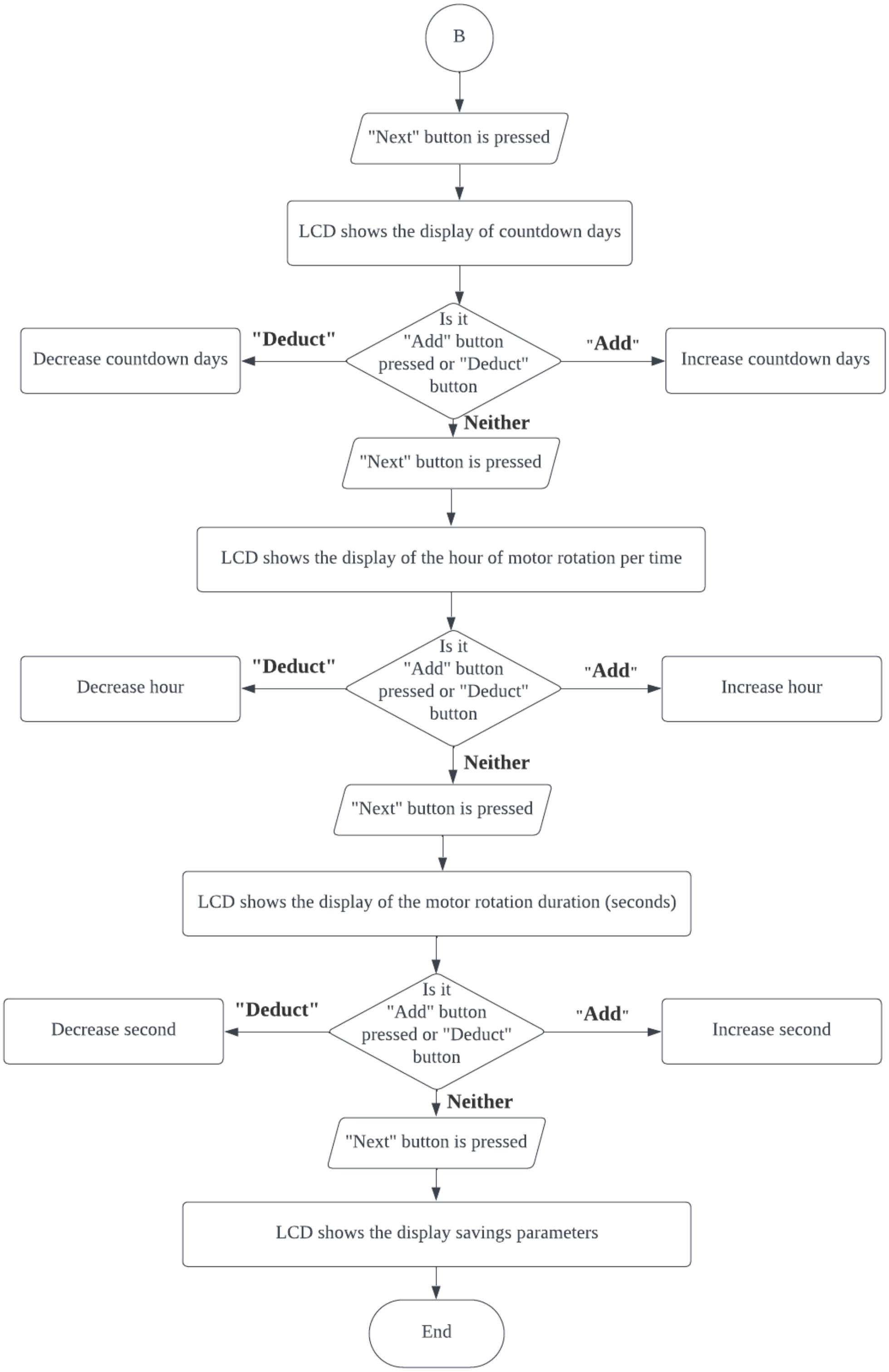


Figure 25: Egg incubator parameters setting flowchart

5.6 EGG INCUBATOR GRAPHICAL USER INTERPHASE FLOWCHART

The flow chart below shows how our egg incubator takes care of the eggs according to the sensor (DHT22), countdown timer and countdown days.

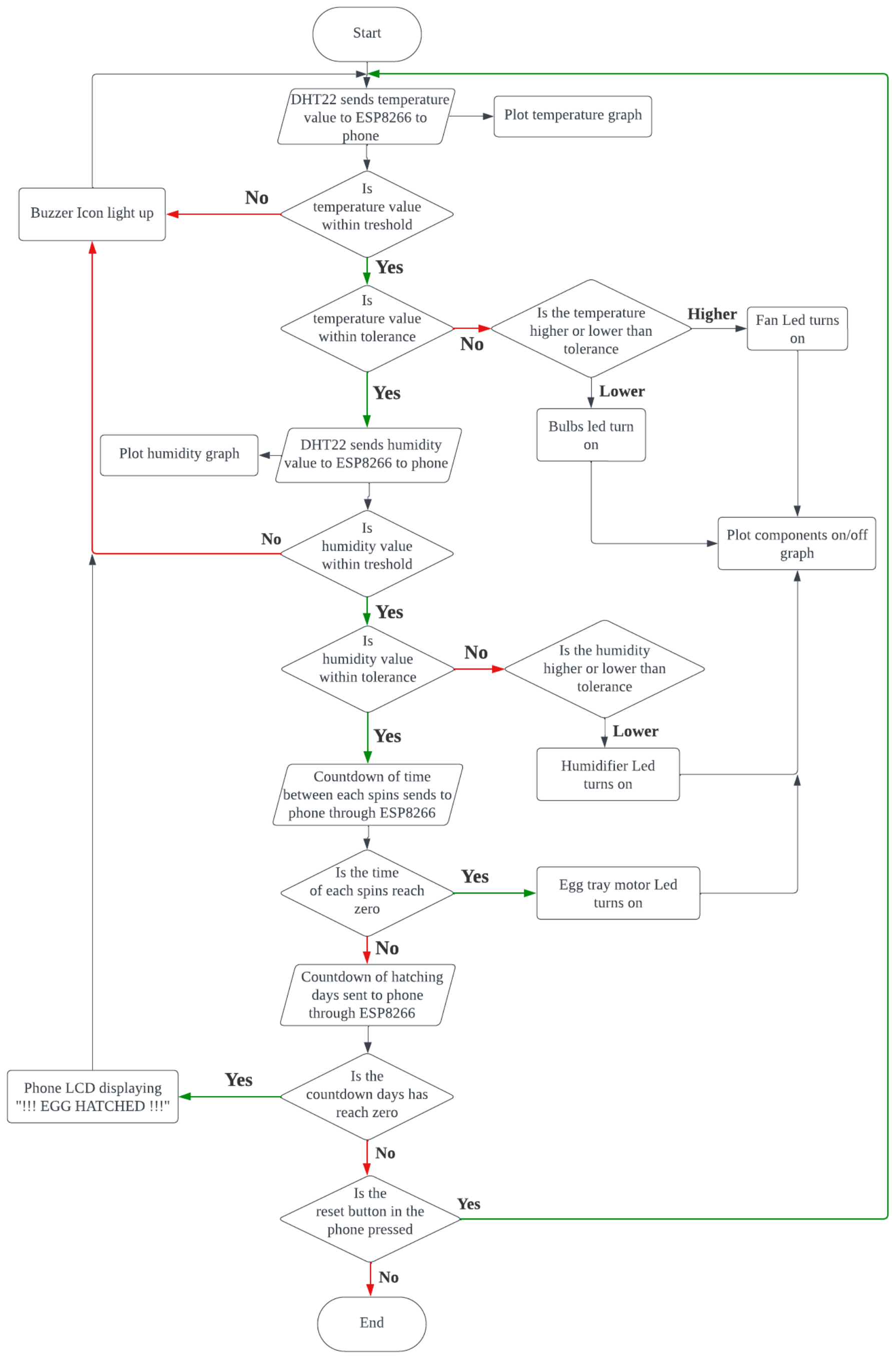


Figure 26: Egg Incubator Graphical User Interphase Flowchart

5.7 GRAPHICAL USER INTERFACE (GUI) DESIGN

Each time the DHT22 sends the data, it will be sent to the phone through ESP8266. The temperature and humidity graph has been plotted on the GUI of our phone. The values of the temperature and humidity would also be displayed on it, same goes for the countdown days and countdown time for each rotation. Each part has been turned on or off, it will be recorded on a graph on our GUI and there is an LCD displaying the egg still not yet hatched as the countdown days have not reached zero yet. If the buzzer is turned on, the icon at the top left corner will be turned on. If the reset button at the top right corner has been pressed, it will trigger the Arduino uno to make it reset and restart, this is for the purpose of preventing the machine from lagging.

Figure 27: Graphical User Interface (GUI) Design

5.8 CODES AND PROTEUS CIRCUIT

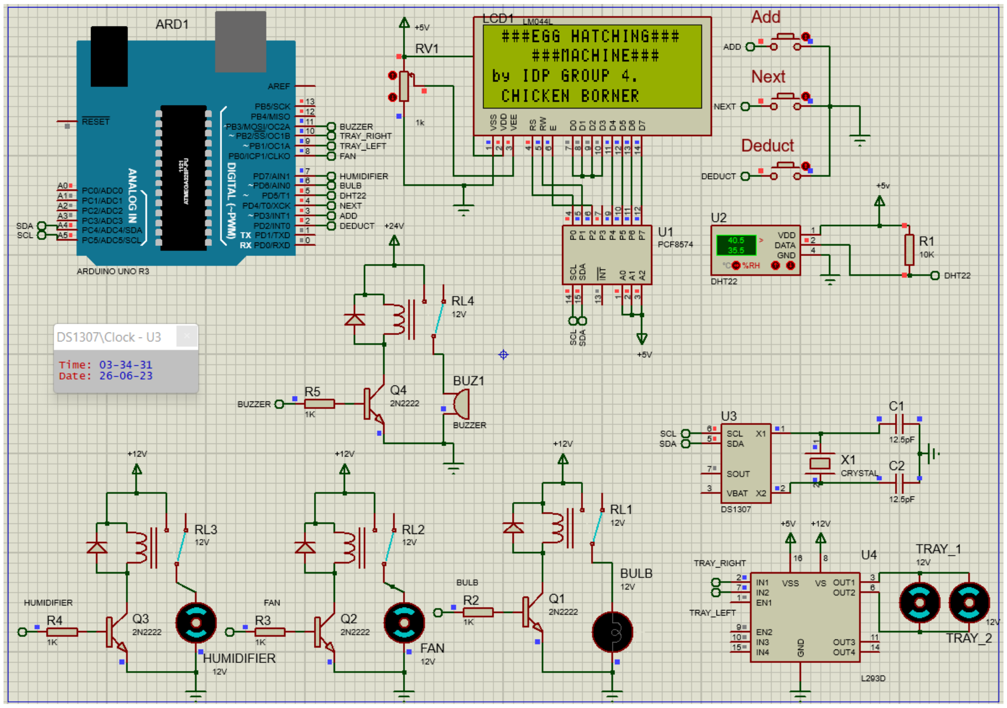


Figure 28: Codes and proteus circuit

The circuit from the figure above is where we simulate our machine, the software is called Proteus. After the code has been compile, there will be a hex file created which we will need to put that hex file into Arduino Uno from the figure above so that we can run our machine virtually to test and run before uploading to the real machine. The codes have also been typed according to the circuit Proteus above.

Due to the codes is too long, it has been uploaded to google drive in the link below:
https://drive.google.com/drive/folders/1Koq8gFaozONAf8DKZPLm2jem8ofWfgHk?usp=sharing

In Arduino IDE, the main libraries that we need to install are stated below:

1. Adafruit BusIO by Adafruit
2. Adafruit Unified Sensor by Adafruit
3. DHT sensor library (version 1.4.4)

4. Liquid Crystal I2C by Frank de Brabander
5. RTClib by Adafruit
6. TimerOne by Paul Stoffregen
7. TinyDHT sensor library by Adafruit

CHAPTER 6

MANUFACTURING

6.1 INTRODUCTION

This chapter focuses on the design and operation of integrated systems to produce high-quality, cost-effective goods. The responsibilities of a manufacturing engineer involve designing complete production processes, establishing tools and equipment, as well as arranging systems and product characteristics to provide the best processes, products, and value of capital investment for each project.

6.2 FABRICATION PROCESS PLANNING PROCESS PLANNING FLOWCHART

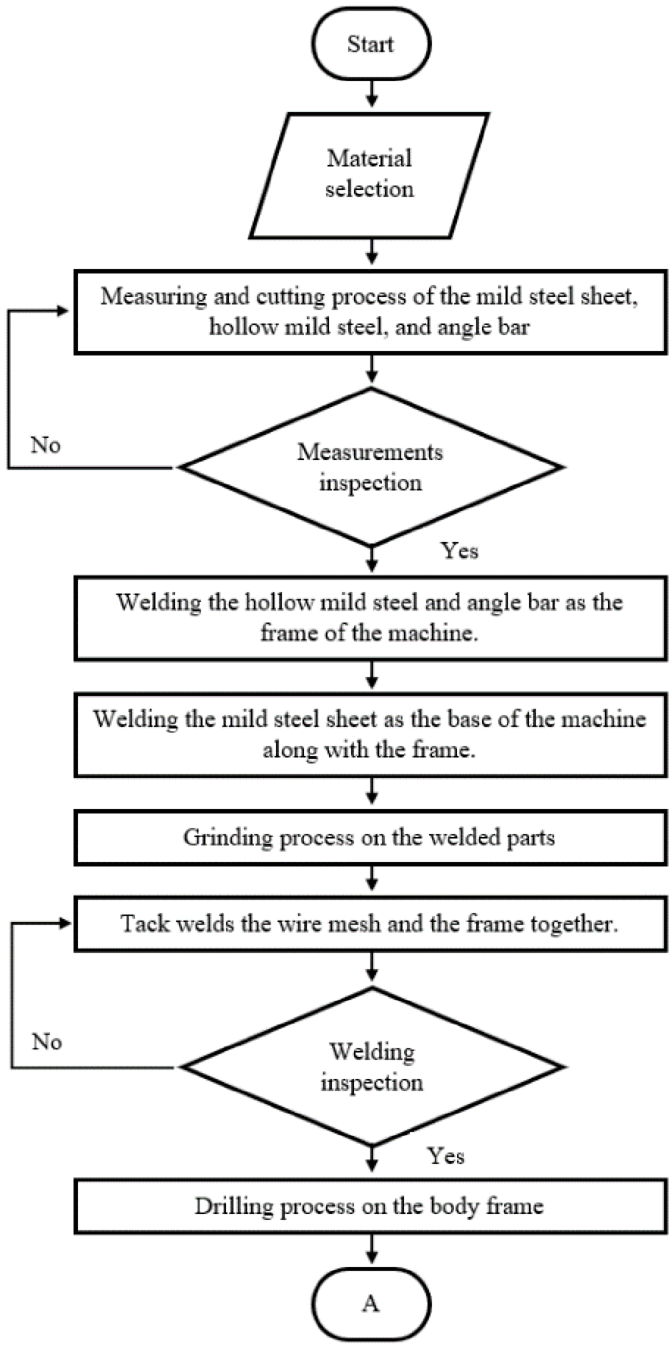


Figure 29: Fabrication Process Planning Process Planning Flowchart

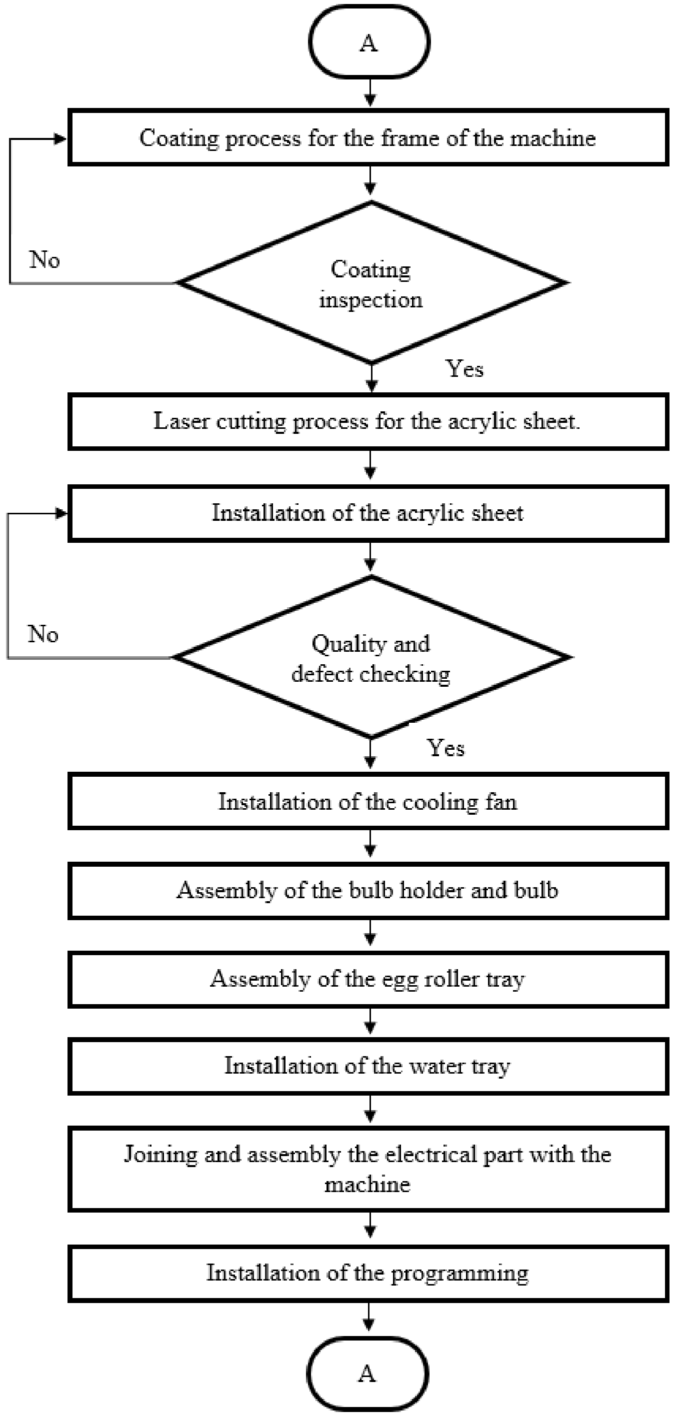


Figure 30: Fabrication Process Planning Process Planning Flowchart

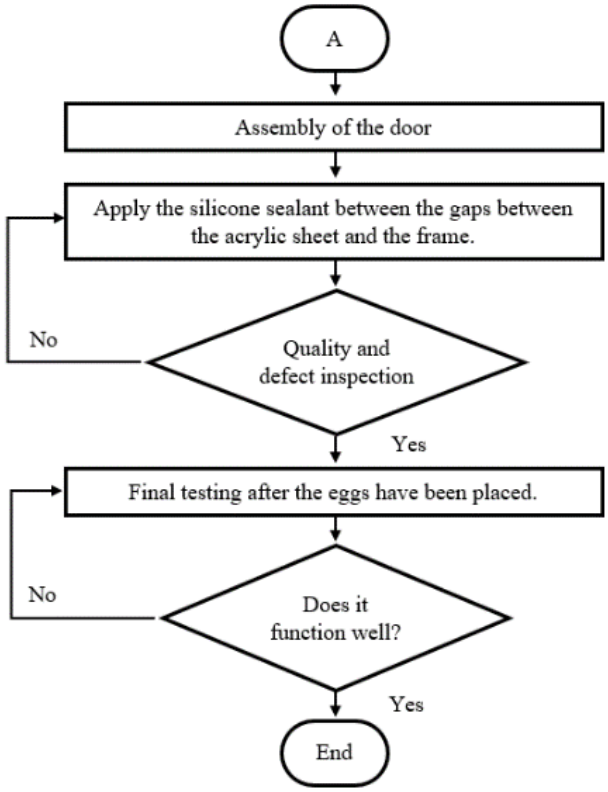


Figure 31: Fabrication Process Planning Process Planning Flowchart

6.3 DETAILS DESCRIPTION ON THE COMPONENT FABRICATION

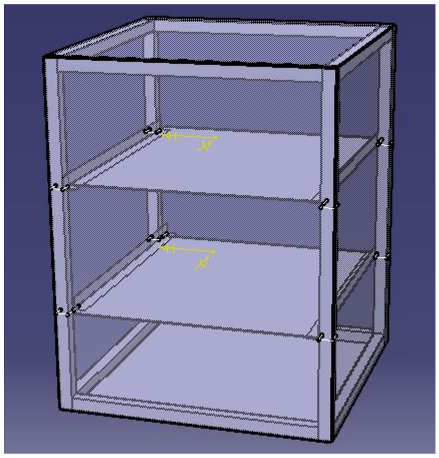


Figure 32: The isometric view of the body frame

6.3.1 PROCESS PLAN OF BODY FRAME

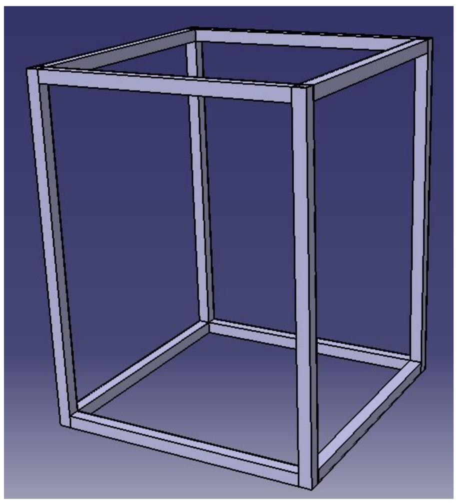
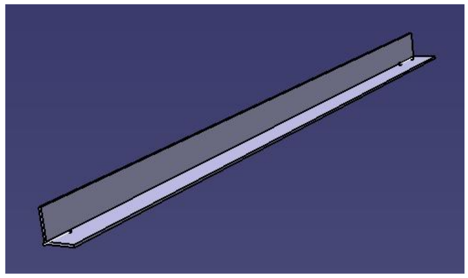
<table>
<tr><td colspan="3">Process plan
</td></tr>
<tr><td colspan="2">Part: Body Frame (Full view)</td><td>Material: Hollow mild steel</td></tr>
<tr><td colspan="2">Project: Egg Hatching Machine</td><td>Size (mm): 500(L)×25(W)×25(H)
Qty: 8
Size (mm): 700(L)×25(W)×25(H)
Qty: 4</td></tr>
<tr><td>No.</td><td>Operational Description</td><td>Tooling/Machine</td></tr>
<tr><td>1.</td><td>Cutting</td><td>Cut off saw</td></tr>
<tr><td>2.</td><td>Joining</td><td>MIG Welding</td></tr>
</table>

Table 11: Process plan of the body frame

6.3.2 PROCESS PLAN OF WIRE MESH HOLDER

Process plan

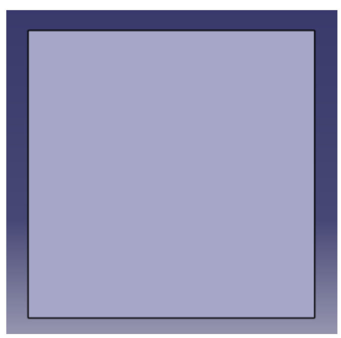


Part: Wire mesh holder		Material: Mild Steel Angle Bar (25×25mm)
Project: Egg Hatching Machine		Size (mm): 500(L)×2(W)×27(H) Qty: 4
No.	Operational Description	Tooling/Machine
1.	Cutting	Cut off saw, Hand drill
2.	Joining	MIG Welding

Table 12: Process plan of wire mesh holder

6.3.3 PROCESS PLAN OF BASE

Process plan



Part: Body (Bottom view)		Material: Sheet metal mild steel
Project: Egg Hatching Machine		Size (mm): 550(L)×2(W)×550(H) Qty: 1
No.	Operational Description	Tooling/Machine
1.	Cutting	Plate cutting machine
2.	Joining	MIG Welding

Table 13: Process plan of the base

6.3.4 PROCESS PLAN OF BODY (SIDE VIEW)

Process plan

Part: Body (Side view)		Material: Acrylic sheet
Project: Egg Hatching Machine		Size (mm): 550(L)×3(W)×700(H) Qty: 4
No.	Operational Description	Tooling/Machine
1.	Cutting	Laser cutting machine
2.	Joining	Hand drill, Bolt, and Nut

Table 14: Process plan of the body of the side view

6.3.4.1 PROCESS PLAN OF BODY (TOP VIEW)

Process plan

Part: Body (Side view)		Material: Acrylic sheet
Project: Egg Hatching Machine		Size (mm): 550(L)×3(W)×700(H) Qty: 4
No.	Operational Description	Tooling/Machine
1.	Cutting	Laser cutting machine
2.	Joining	Hand drill, Bolt, and Nut

Table 15: Process plan of the body of the top view

6.3.5 PROCESS PLAN OF EGG TRAY HOLDER

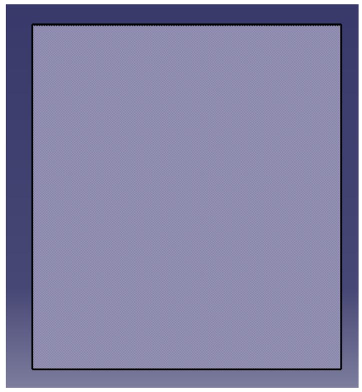
<table>
<tr><td colspan="3">Process plan
</td></tr>
<tr><td colspan="2">Part: Egg tray holder</td><td>Material: Wire mesh</td></tr>
<tr><td colspan="2">Project: Egg Hatching Machine</td><td>Size (mm): 550(L)×3(W)×550(H)
Qty: 2</td></tr>
<tr><td>No.</td><td>Operational Description</td><td>Tooling/Machine</td></tr>
<tr><td>1.</td><td>Cutting</td><td>Plate cutting machine</td></tr>
<tr><td>2.</td><td>Joining</td><td>Tack Welding</td></tr>
</table>

Table 16: Process plan of egg tray holder

6.4 PRODUCT STRUCTURE

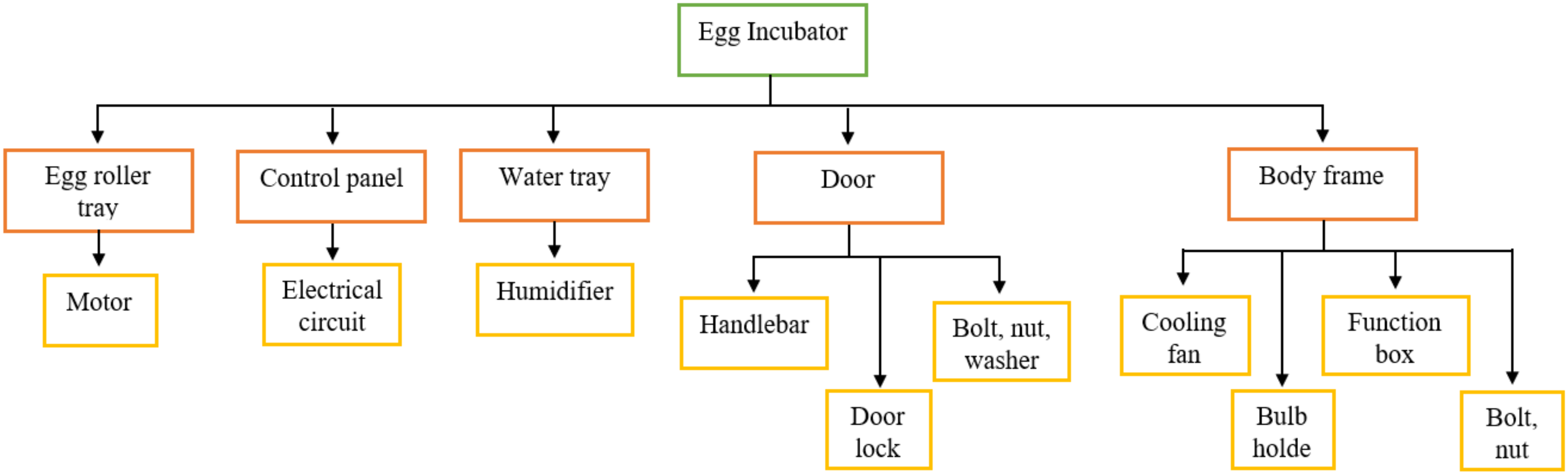


Figure 33: The product structure of the machine

6.5 ASSEMBLY OPERATIONS

STEPS	FIGURE	PROCESS	ASSEMBLY METHODS
1		Measuring and cutting process of the mild steel sheet, hollow mild steel, and angle bar	-
2		Tack welding process for the body frame	Tack welding
3		MIG welding the hollow mild steel, angle bar and mild steel sheet as the frame of the machine	Tack welding
4		Grinding process on the welded parts	-

5		Tack welds the wire mesh and the frame together	Tack welding
6		Coating process for the frame of the machine	-
7		Laser cutting process for the acrylic sheet	-
8		Drilling process on the body frame	By using hand drill
9		Installation of the acrylic sheet with the body frame	Assemble the acrylic sheet with bolts and nuts

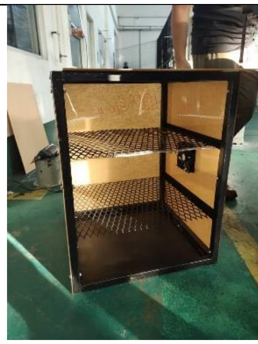
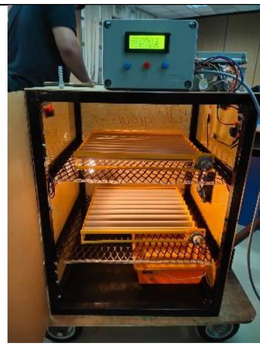
10		Installation of bulb holder and cooling fan	Attach the bulb holder and cooling fan with bolts and nuts
11		Joining and assembly the electrical part with the machine and installation of the programming	-
12		Final testing after the eggs have been placed	Assemble the door using bolts and nuts as the final process

Table 17: The assembly operations

6.6 MANUFACTURING PROCESS PLANNING

6.6.1 BILL OF MATERIALS (BOM) WITH PART NO AND COST PER UNIT

Electrical and Electronic Components					
No	**Components**	**Quantity**	**Price per Unit (RM)**	**Total Price (RM)**	**Source**
1	5VDC Arduino Uno	1		RM0.00	Sponsor
1	Jumper wire Female to Male	1	RM3.70	RM3.70	Shop
2	Jumper wire Female to Female	1	RM3.70	RM3.70	Shop
3	Jumper wire Male to Male	1	RM3.70	RM3.70	Shop
4	DHT22	1	RM14.90	RM14.90	Shop
5	3-24VDC Buzzer	1	RM2.50	RM2.50	Shop
6	LCD Display (20x4)	1	RM19.90	RM19.90	Shop
7	Push button	2	RM1.00	RM2.00	Shop
8	E27 Bulb Holder	2	RM3.38	RM6.76	Shopee
9	E27 Light Bulb	2	RM16.35	RM32.70	Shopee
10	12VDC Cooling fan	1	RM9.90	RM9.90	Shop
11	ESP8266	1	RM19.90	RM19.90	Shop
12	WeMos D1	2	RM22.35	RM44.70	Shopee
13	Junction box	2	RM8.50	RM17.00	Shop
14	Humidifier	2	RM10.01	RM20.01	Shopee
15	12V Relay 4 Ways	1	RM12.90	RM12.90	Shop

16	12V Relay 1 Way	2	RM4.90	RM9.80	Shop
17	DS1307 Tiny RTC Module	1	RM4.90	RM4.90	Shop
18	L293D DC Motor Driver	2	RM14.90	RM29.80	Shop
19	I2C Interface PCF8574	2	RM4.60	RM9.20	Shop
20	1k Ohm Potentiometer	2	RM1.00	RM2.00	Shop
21	Diode 1N4001	6	RM1.00	RM6.00	Shop
22	Resistor 10k Ohm	1	RM1.00	RM1.00	Shop
23	Transistor 2N2222	2	RM1.00	RM2.00	Shop
24	Red Lamp Light Switch	2	RM6.93	RM13.86	Shopee
25	Breadboard (830 holes)	1	RM3.90	RM3.90	Shop
27	Adapter power supply	1		RM0.00	Sponsor
Total cost				RM296.73	

Table 18: The cost of Electrical and Electronics Components

Manufacturing and Mechanical Materials					
No	**Materials**	**Quantity**	**Price per Unit (RM)**	**Total Price (RM)**	**Source**
1	Rectangular plastic tray (H66mm)	1	RM5.30	RM5.30	Shop
2	Bolt, and Nut	3	RM3.90	RM11.70	Shop
3	Hinge	1 set	RM5.50	RM5.50	Shop
4	Door handle	1	RM2.50	RM2.50	Shop

5	Door lock	1	RM2.50	RM2.50	Shop
6	Egg tray with synchronous motor	2	RM65.80	RM131.60	Shopee
7	Silicone sealant	1	RM20.00	RM20.00	Shop
8	Acrylic sheet (A2, Thickness 5mm, 400x175)			RM0.00	Sponsor
9	Wire mesh (300x300mm)			RM0.00	Sponsor
10	Hollow mild steel			RM0.00	Sponsor
11	Mild steel angle bars			RM0.00	Sponsor
Total cost				RM179.10	

Table 19: The cost of Manufacturing and Mechanical Components

No.	Bill of Materials	Cost Summation (RM)
1.	Electrical and Electronic Components	RM296.73
2.	Manufacturing and Mechanical materials	RM179.10
Total Cost Summation (RM)		RM475.83

Table 20: The overall cost of developing the machine.

6.6.2 PROCESS FLOW (PRECEDENCE DIAGRAM) AND CYCLE TIME

The Precedence Diagram Method (PDM) is a visual representation technique depicting a project's activities. It is a method for constructing a network diagram of a project schedule that uses boxes or nodes to represent activities and connects them with arrows to demonstrate dependencies. The Precedence Diagram Method tries to produce a more precise scheduling network diagram.

Work Element	**Description**	**Predecessor**	**Operating time (min)**
A	Material selection, measuring and cutting the material	-	10
B	Welding and grinding process	A	20
C	Coating process	B	20
D	Laser cutting the acrylic sheet	C	10
E	Drilling process on the body frame and assemble with bolt and nut	D	15
F	Assembly of the bulb holder, cooling fan, egg roller tray and water tray	E	05
G	Develop the electrical circuit	F	15
H	Assemble the circuit in the machine	F	15
I	Programming installation	G, H	15
J	Quality inspection	I	05
K	Packaging	J	10
TOTAL WORK			140

Table 21: The precedence time with the cycle time.

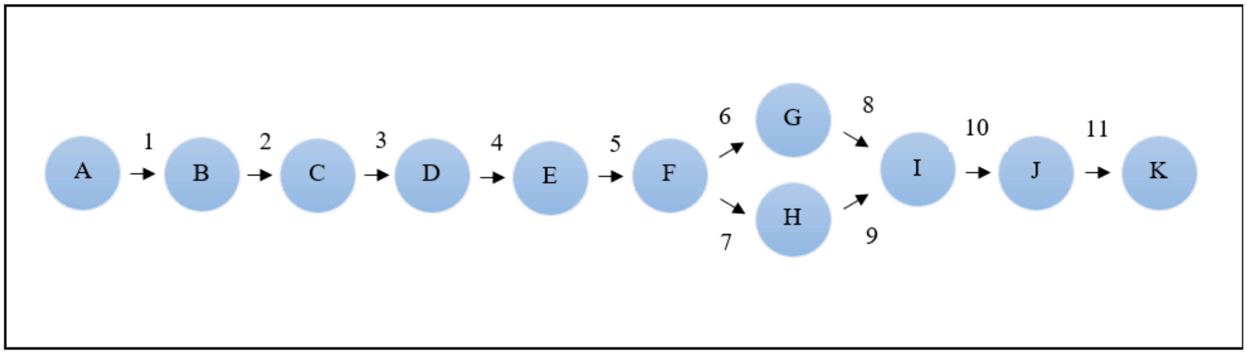


Figure 34: Precedence diagram of the process flow

The table above shows the procedures that will be completed to produce the machine. A precedence diagram will be generated from the database to represent the network. According to the precedence diagram, there is one self-sustaining activity that can begin without waiting for the others to finish. The product will be tested and inspected to ensure that the mechanical and electrical components may interact with each other using software. Finally, the products will be packaged before being shipped to the consumer.

6.6.3 LINE SIMULATION AND FACTORY LAYOUT

A manufacturer can use the Tecnomatix suite of digital manufacturing technologies to digitize manufacturing and the process of converting raw materials into finished goods. Tecnomatix software supports synchronizing product engineering, manufacturing engineering, production, and service processes to improve production efficiency and actualize innovation.

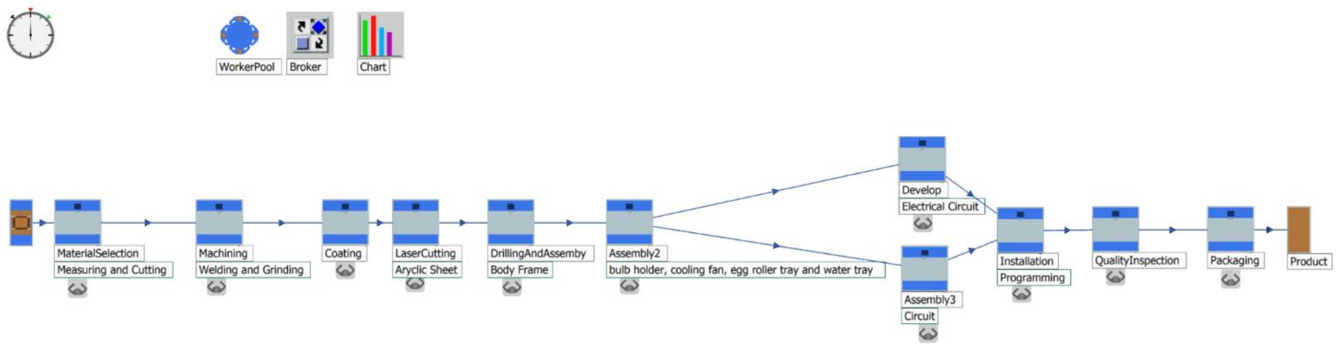


Figure 35: The Line Simulation using Tecnomatix Plant Simulation

The figure above depicts a production line simulation utilizing Tecnomatix software to create our product, an egg incubator. We assigned a person to each workstation to do their duties, from preparation to packaging. Each station has two or one worker to do their assignment. The workers are divided according to the time it takes to complete the job at each station.

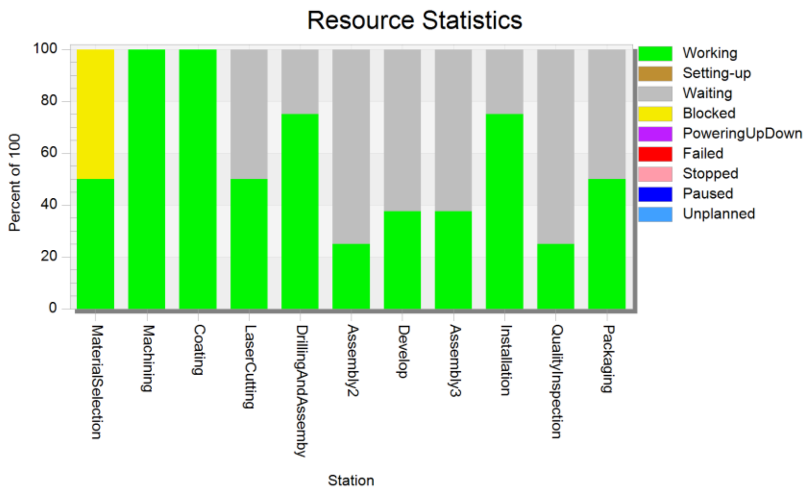


Figure 36: The Resource Statistics

Figure 1 depicts the statistical resources from the line simulation. Resource use statistics are compiled to document the performance and operational data of the resources consumed by integration servers. According to the statistics, assembly stations for the bulb holder, cooling fan, egg roller tray, and water tray have a longer wait time than the drilling procedure on the body frame and assembly with bolt and nut. This is because the drilling process on the body frame and assembly with the bolt and nut department must be completed before delivering the station assembly for the bulb holder, cooling fan, egg roller tray, and water tray.

Then, the yellow color shows those workstations must remain idle until the subsequent workstations have completed its task before they can operate and resume machine production. Finally, the station of preparation is higher for waiting since it has one worker and requires assistance from the station to complete the task after the one product on preparation has already been completed.

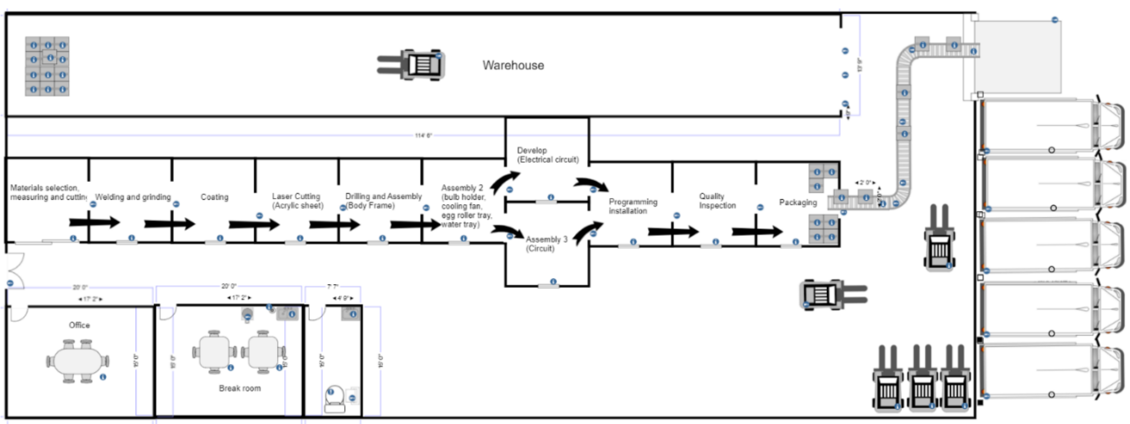


Figure 37: The Factory Layout

The arrangement of equipment, workspaces, and support areas within a factory is referred to as plant layout. Plant layout comprises the construction of physical links between buildings, machinery, and industrial procedures to ensure that the manufacturing process is carried out successfully.

Figure displays the layout of a factory as well as the support sections where products are built before being transported to customers. The production system is set up to provide as smooth a continuous flow line as possible by grouping the workstations into lots and classifying them by the manufacturing process and machining sequences. Before beginning the assembly process, the raw material is measured, cut into the desired dimension, welded, and drilled. After that, a part must be delivered to a welding station before starting process drilling. The machine is then constructed with all its machining process pieces, and it is then fitted together using electrical and electronic parts that have been connected from other lots. To make certain that the product is delivered safely to the customer, it will be programmed and checked for reliability before being packaged.

6.6.4 COST ANALYSIS (OPERATION AND COST PER UNIT)

Working:	80.65%	Average lifespan:	2:34:59.9548
Setting-up:	0.00%	Average exit interval:	19:59.9973
Waiting:	19.35%	Total throughput:	48163
Stopped:	0.00%	Throughput per minute:	0.05
Failed:	0.00%	Throughput per hour:	3.00
Paused:	0.00%	Throughput per day:	71.99

Figure 38: The cost analysis of the machine

Calculation Analysis

Working hours = 8 hours/day

Working days = 5 days/week

Weeks per year = 52 weeks

1. Daily production rate = Throughput per hours × Working hours

   = 3.00 units × 8 hours

   = 24 unit per day

2. Demand = Throughput per hour × working hours × working days × weeks per year

   = 3.00 units × 8 hours × 5 days × 52 weeks

   = 6240 unit per year

3. Total work content (Twc)

= 10 + 20 + 20 + 10 + 15 + 05 + 15 + 15 + 15 + 05 + 10

= 140 minutes

4. Cycle time, Tc = Available time / unit per hour

   = 60 / 3.00

   = 20 minute per hours

5. Number of workstations, N = Twc / Tc

= 140 / 20

= 7 Workstations

Cost Analysis

Working hours	= 8 hours/day
Working day	= 5 days/week
Working weeks	= 52 weeks/year
Labor salary	= RM1800/month
Total workers	= 13 workers
Annual Demand	= 6240 unit per year

Labor cost per year

= Labor salary × Total workers × Time required (year)

= RM1800 × 11 × 12 years

= RM 237,600 per year per labor

Production Cost

Fixed cost				
No	Item	Quantity	Unit cost (RM)	Total Cost (RM)
1.	Premise Rental	1 unit × 12 months	4500	54000
2.	Machine Rental	5 units × 12 months	300	18000
3.	Utilities	1 unit × 12 months	900	10800
4.	App Development Cost	1 unit	500	500
5.	Miscellaneous (PPE)	12 units × 12 months	30	4320
6.	Insurance (Warranty)	6240 units	166	1,035,840
Total fixed cost/year				1,123,460
Total fixed cost/year				180.04

Table 22: The fixed cost analysis

Variable cost				
No	Item	Quantity	Unit cost (RM)	Total Cost (RM)
1.	Direct labour	11 pax × 12 months	1800	237,600
2.	Transportation	6240 units	4.50	28080
3.	Shipping	6240 units	7.50	46800
4.	Material Cost	6240 units	475.83	2,969,179.20
Total variable cost/year				3,281,659.20
Total variable cost/unit				525.90

Table 23: The variable cost analysis

Production Cost		
Cost	Per year (RM)	Per unit (RM)
Total fixed cost	1,123,460	180.04
Total variable cost	3,281,659.20	525.90
Production cost	4,405,119.20	705.94

Table 24: The production cost analysis

Selling price per unit

The selling price is how much a buyer pays for a product or service.

Mark up 35%

Selling price (SP)

= Production cost for units × 1.35

= RM705.94 × 1.35

= RM953.02

= RM1100

Therefore, the selling price for a mark-up of 35% is RM 953.02 but we convert into RM 1100 of selling price.

6.4 Break Even Point (BEP) Analysis

The break-even point is the point at which revenue and costs are equal. A break-even analysis is a financial computation that compares the costs of a new business, service, or product to the unit sale price to calculate when we will be profitable. In other words, it indicates when you will have sold enough units to cover all our expenses.

1. Total Revenue

= Annual demand × Selling price per unit

= 6240 units × RM1100

= RM6,864,000

2. Contribution Margin

= Selling price - Total variable cost per unit

= RM1100 - RM50

= RM1050

3. Gross margin

= 6240 units × RM1050

= RM 6,552,000

4. Net Profit/Loss

= Gross margin - Total production cost

= RM 6,552,000 - RM 4,405,119.20

= RM 2,146,880.80

5. BEP

= Total fixed costs / Contribution margin per unit

= RM 4,405,119.20 / RM1100

= 4004.65 units

= 4005 units

Unit sold	Fixed Cost (RM)	Variable Cost (RM)	Total Cost (RM)	Total Sales (RM)	Profit (RM)
0	1,123,460.00		1,123,460.00	0	2,099,784.00
560	1,123,460.00	2850.00	1,126,310.00	307,610.00	1,795,024.00
1120	1,123,460.00	5700.00	1,129,160.00	615,220.00	1,490,264.00
1680	1,123,460.00	8550.00	1,132,010.00	922,830.00	1,185,504.00
2240	1,123,460.00	11400.00	1,134,860.00	1,230,440.00	880,744.00
2800	1,123,460.00	14250.00	1,137,710.00	1,538,050.00	575,984.00
3360	1,123,460.00	17100.00	1,140,560.00	1,845,660.00	271,224.00
3920	1,123,460.00	19950.00	1,143,410.00	2,153,270.00	33,536.00
4480	1,123,460.00	22800.00	1,146,260.00	2,460,880.00	338,296.00
5040	1,123,460.00	25650.00	1,149,110.00	2,786,490.00	643,056.00
5600	1,123,460.00	28500.00	1,151,960.00	3,076,100.00	947,816.00
6240	1,123,460.00	313500.00	1,154,810.00	3,256,790.00	1,286,112.00

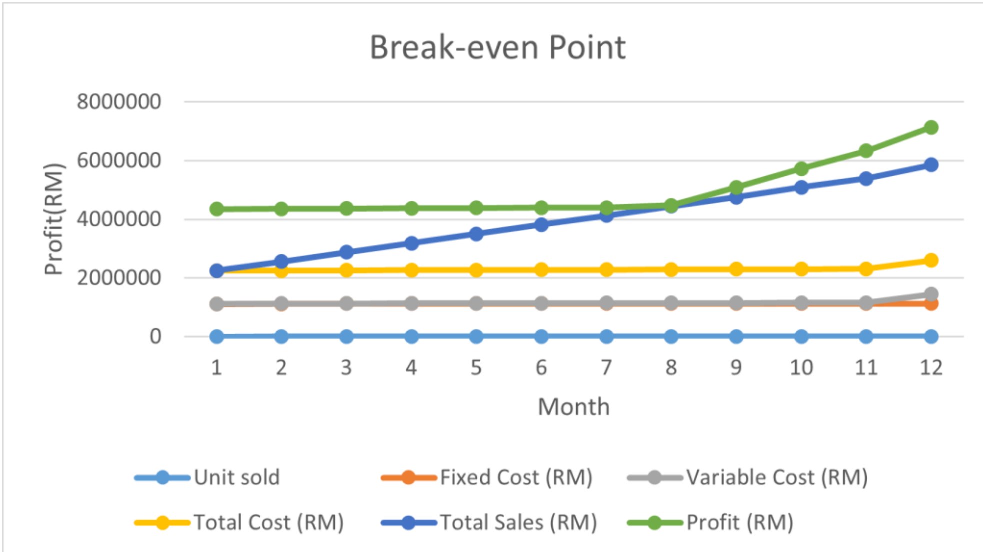


Figure 39: The Break-even point.

Then, after we know the break-even point analysis, we need to know the return of investment (ROI). Return on investment (ROI) is a performance measure used to evaluate an investment's efficiency or profitability or compare the efficiency of many different investments.

Hence, ROI formula is:

ROI = [(Investment Gains – Investment Costs)/ Investment Costs) x 100]

Return of Investment (ROI)

= [(RM6,864,000.00 - RM 4,405,119.20) / (RM 4,405,119.20)] x 100%

= 55.81 %

CHAPTER 7

PROCUREMENT ANALYSIS

7.1 PROTOTYPE COMPONENT SOURCES AND COST

Electrical and Electronic Components					
No	**Components**	**Quantity**	**Price per Unit (RM)**	**Total Price (RM)**	**Source**
1	5VDC Arduino Uno	1		RM0.00	Sponsor
1	Jumper wire Female to Male	1	RM3.70	RM3.70	Shop
2	Jumper wire Female to Female	1	RM3.70	RM3.70	Shop
3	Jumper wire Male to Male	1	RM3.70	RM3.70	Shop
4	DHT22	1	RM14.90	RM14.90	Shop
5	3-24VDC Buzzer	1	RM2.50	RM2.50	Shop
6	LCD Display (20x4)	1	RM19.90	RM19.90	Shop
7	Push button	2	RM1.00	RM2.00	Shop
8	E27 Bulb Holder	2	RM3.38	RM6.76	Shopee
9	E27 Light Bulb	2	RM16.35	RM32.70	Shopee
10	12VDC Cooling fan	1	RM9.90	RM9.90	Shop
11	ESP8266	1	RM19.90	RM19.90	Shop
12	WeMos D1	2	RM22.35	RM44.70	Shopee
13	Junction box	2	RM8.50	RM17.00	Shop

14	Humidifier	2	RM10.01	RM20.01	Shopee
15	12V Relay 4 Ways	1	RM12.90	RM12.90	Shop
16	12V Relay 1 Way	2	RM4.90	RM9.80	Shop
17	DS1307 Tiny RTC Module	1	RM4.90	RM4.90	Shop
18	L293D DC Motor Driver	2	RM14.90	RM29.80	Shop
19	I2C Interface PCF8574	2	RM4.60	RM9.20	Shop
20	1k Ohm Potentiometer	2	RM1.00	RM2.00	Shop
21	Diode 1N4001	6	RM1.00	RM6.00	Shop
22	Resistor 10k Ohm	1	RM1.00	RM1.00	Shop
23	Transistor 2N2222	2	RM1.00	RM2.00	Shop
24	Red Lamp Light Switch	2	RM6.93	RM13.86	Shopee
25	Breadboard (830 holes)	1	RM3.90	RM3.90	Shop
27	Adapter power supply	1		RM0.00	Sponsor
Total cost				RM296.73	

Table 25: The cost of Electrical and Electronics Components

Manufacturing and Mechanical Materials					
No	**Materials**	**Quantity**	**Price per Unit (RM)**	**Total Price (RM)**	**Source**
1	Rectangular plastic tray (H66mm)	1	RM5.30	RM5.30	Shop
2	Bolt, and Nut	3	RM3.90	RM11.70	Shop
3	Hinge	1 set	RM5.50	RM5.50	Shop
4	Door handle	1	RM2.50	RM2.50	Shop
5	Door lock	1	RM2.50	RM2.50	Shop
6	Egg tray with synchronous motor	2	RM65.80	RM131.60	Shopee
7	Silicone sealant	1	RM20.00	RM20.00	Shop
8	Acrylic sheet (A2, Thickness 5mm, 400x175)			RM0.00	Sponsor
9	Wire mesh (300x300mm)			RM0.00	Sponsor
10	Hollow mild steel			RM0.00	Sponsor
11	Mild steel angle bars			RM0.00	Sponsor
Total cost				RM179.10	

Table 26: The cost of Manufacturing and Mechanical Components

No.	Bill of Materials	Cost Summation (RM)
1.	Electrical and Electronic Components	RM296.73
2.	Manufacturing and Mechanical materials	RM179.10
Total Cost Summation (RM)		RM475.83

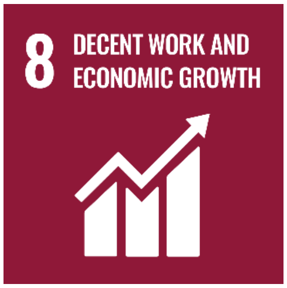
Table 27: The overall cost of developing the machine.

7.2 ENVIRONMENT AND SUSTAINABILITY DISCUSSION

Goal 2: Zero Hunger: By providing a controlled environment for incubating eggs, the incubator can contribute to increasing food production. Eggs are a rich source of protein and vital nutrients, and their successful incubation can lead to the production of more chickens for meat and egg production, thereby enhancing food security and reducing hunger.

Goal 3: Good Health and Well-being: Improved egg incubators can help prevent the spread of diseases by maintaining optimal conditions for egg development. They can minimize the risk of contamination and disease transmission, resulting in healthier chicks and improved overall animal health.

Goal 7: Affordable and Clean Energy: Egg incubators often require a consistent and reliable source of energy for temperature control. Utilizing renewable energy sources, such as solar power, for powering incubators can contribute to the adoption of clean energy solutions, reducing greenhouse gas emissions and promoting sustainable energy practices.



Goal 8: Decent Work and Economic Growth: Egg incubators can support small-scale farmers and entrepreneurs by enabling them to expand their poultry operations. By improving the hatchability of eggs and increasing the survival rate of chicks, incubators can enhance productivity and income opportunities, thereby promoting economic growth and creating employment.

Goal 12: Responsible Consumption and Production: Advanced egg incubators can optimize resource utilization, such as energy and water, by incorporating energy-efficient technologies and minimizing waste. This supports sustainable production practices and encourages responsible consumption of resources.

Goal 13: Climate Action: Egg incubators that utilize energy-efficient technologies and cleaner energy sources help reduce carbon emissions and mitigate the impact of climate change. Additionally, by improving poultry production efficiency, they can contribute to reducing the environmental footprint associated with traditional farming practices.

Goal 15: Life on Land: Egg incubators can play a role in conservation efforts by supporting the breeding and preservation of endangered or threatened species. Controlled breeding programs can aid in the restoration and conservation of various bird species, contributing to biodiversity preservation.

7.3 MATERIALS AND DISPOSAL

Mild Steel and Steel

These materials are commonly used for the structural framework and components of the incubator. After the end of the incubator's life cycle, mild steel and steel components can be recycled. Recycling steel helps conserve natural resources and reduce energy consumption compared to producing new steel from raw materials. It is important to properly segregate and dispose of these materials at recycling centers or through appropriate waste management systems to ensure they are recycled effectively.

Acrylic Sheet

Acrylic sheets are often used for the transparent panels or windows of the egg incubator. Acrylic is a recyclable material, and like steel, it should be separated from other waste and taken to recycling facilities. If the acrylic sheets are still in good condition, they can be reused or donated to other projects or individuals who may find them useful. If they are damaged or no longer usable, recycling is the preferred disposal method.

Electric Components

Electric components, such as wiring, switches, sensors, and heating elements, are crucial for the functioning of the incubator. It is important to dispose of these components in an environmentally responsible manner. In many areas, there are specific collection centers or recycling programs for electronic waste (e-waste). These centers ensure that hazardous substances are properly handled and disposed of, while valuable materials are recovered and recycled. Contact local recycling or e-waste management facilities to determine the appropriate disposal method for these components.

7.4 ENERGY CONSUMPTION

Energy consumption is an important aspect to consider when evaluating the sustainability of an egg incubator.

Energy Efficiency

It is essential to design the egg incubator with energy efficiency in mind. This involves using energy-efficient components, such as LED lights or low-power heating elements and optimizing insulation to minimize heat loss. By reducing energy wastage, an energy-efficient incubator can lower operational costs and decrease its overall environmental impact.

Power Source

The choice of power source for the egg incubator can also influence its energy consumption. Whenever possible, utilizing renewable energy sources, such as solar panels, can significantly reduce reliance on fossil fuels and contribute to a cleaner and more sustainable energy supply. Solar-powered incubators can operate off-grid, making them particularly suitable for rural areas with limited access to electricity.

Automation and Controls

Incorporating automated controls and sensors in the egg incubator can help optimize energy consumption. These features allow for precise monitoring and adjustment of temperature, humidity, and ventilation, ensuring that the incubator operates within the required parameters while minimizing energy usage. Additionally, using timers or programmable settings can further enhance energy efficiency by operating the incubator only when necessary.

Maintenance and Calibration

Regular maintenance and calibration of the egg incubator's components are crucial to ensure optimal energy performance. Malfunctioning or poorly calibrated equipment may lead to energy inefficiencies or unnecessary energy consumption. Regular inspections, cleaning, and proper maintenance of electrical connections and heating elements can help identify and address any issues that may impact energy efficiency.

Education and Awareness

Promoting energy-conscious practices among users of the egg incubator can help reduce energy consumption. Providing guidelines and information on energy-saving tips, such as avoiding unnecessary opening of the incubator, can encourage users to make conscious choices that minimize energy waste.

7.5 PRODUCT LIFECYCLE

Considering the product lifecycle of an egg incubator is crucial for understanding its overall sustainability. The lifecycle typically consists of several stages as stated below:

Stage 1: Raw Material Extraction

The first stage involves sourcing the raw materials required to manufacture the incubator, such as mild steel, steel, acrylic sheets, and electronic components. Sustainable practices in this stage may involve using recycled or responsibly sourced materials to minimize environmental impact.

Stage 2: Manufacturing and Assembly

The manufacturing and assembly process of the egg incubator can have environmental and social implications. Adopting energy-efficient manufacturing practices, minimizing waste generation, and ensuring safe working conditions for the workers involved are important considerations in this stage.

Stage 3: Distribution and Transportation

The transportation of the incubator from the manufacturing facility to the end-users contributes to its carbon footprint. Optimizing transportation routes, using efficient packaging materials, and exploring greener transportation options can help reduce emissions associated with distribution.

Stage 4: Product Us

The use phase is where the egg incubator serves its primary function of incubating eggs. Sustainable design features, such as energy-efficient components, insulation, and automation, can minimize energy consumption and optimize performance. Providing clear instructions and guidelines for efficient use can also contribute to the product's sustainability during this phase.

Stage 5: Maintenance and Repair

Proper maintenance and repair practices prolong the lifespan of the incubator, reducing the need for frequent replacements. Providing users with information on maintenance routines, troubleshooting, and access to spare parts can help extend the product's life and reduce waste generation.

Stage 6: End of Life

When the egg incubator reaches the end of its usable life, proper disposal or recycling is important. Recycling materials like steel, acrylic, and electronic components can recover valuable resources while minimizing environmental impact. Designing the product with modularity and ease of disassembly in mind can facilitate recycling and responsible disposal.

7.6 PROJECT PLANNING AND PROGRESS

It is essential to understand the entire purchasing and planning process. The flow is critical to avoiding time wasted dealing with the required process component. Dealing with vendors who manage group spending also allows us to avoid costly purchases. To address the issue, purchasing flowcharts and a procurement strategy were developed to help our team negotiate the best-planned deal for us.

CHAPTER 8
RELIABILITY TESTING

8.1 INTRODUCTION

To test whether the egg incubator can adapt in various environments. We have placed our machine into three different kinds of environments which is room with air conditioning, room with normal temperature, and outside of room.

We will record a few parameters which are the time taken for the temperature to increase when the bulb is turned on, time taken for the humidity to increase when the bulb is turned on, time taken for the temperature to drop when the fan is turned on.

8.2 TESTING OF THE MACHINE WHEN IT IS OPERATING 24 HOURS EVERYDAY

After we have done the assembly of the mechanical and electrical parts of the egg incubator, we left it on for every single day in each environment (cold, room and hot) to test and analyze whether our egg incubator will experienced any malfunction and deteriorate.

Although we experience some lagging and malfunctions of DHT22, we have solved the problem by modifying the Arduino Codes for the machine to self-reset or refresh as we found out that the machine had experienced issues due to its memory has been store fully which causing the machine to lag or stop operating.

8.3 GRAPH OF THE EGG INCUBATOR IN EACH ENVIRONMENT

8.3.1 EGG INCUBATOR IN A ROOM WITH AIR CONDITIONING

Figure 40: The egg incubator has been placed into a room with air conditioning.

TEMPERATURE VS TIME GRAPH FOR BULBS BEING TURNED ON

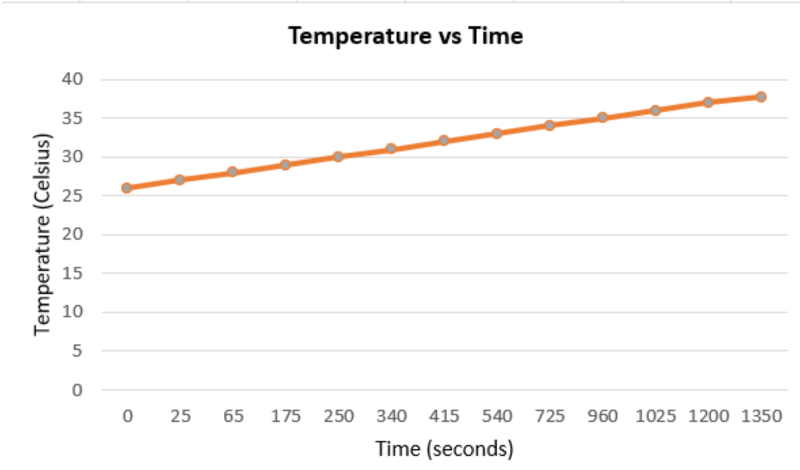


figure 41: Temperature VS Time Graph for Bulbs Being Turned On

HUMIDITY VS TIME GRAPH FOR HUMIDIFIER BEING TURNED ON

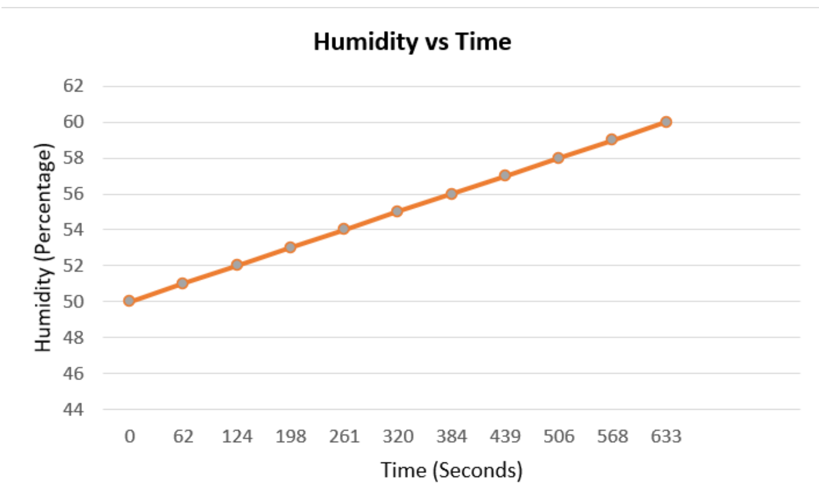


Figure 42: Humidity Vs Time Graph for Humidifier Being Turned On

TEMPERATURE VS TIME GRAPH FOR FAN BEING TURNED ON

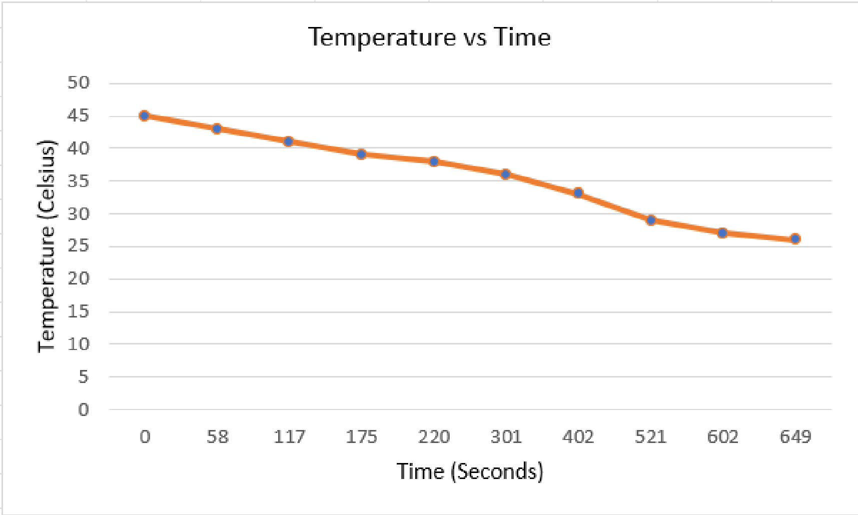


Figure 43: Temperature Vs Time Graph for Fan Being Turned On

EGG INCUBATOR IN A ROOM WITH ROOM TEMPERATURE

Figure 44: The egg incubator has been placed into a room with normal temperature.

TEMPERATURE VS TIME GRAPH FOR BULBS BEING TURNED ON

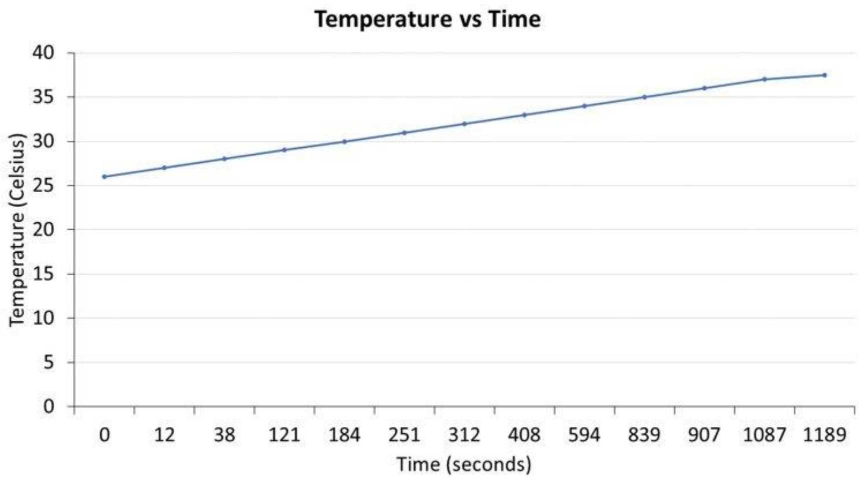


Figure 45: Temperature Vs Time Graph for Bulbs Being Turned On

HUMIDITY VS TIME GRAPH FOR HUMIDIFIER BEING TURNED ON

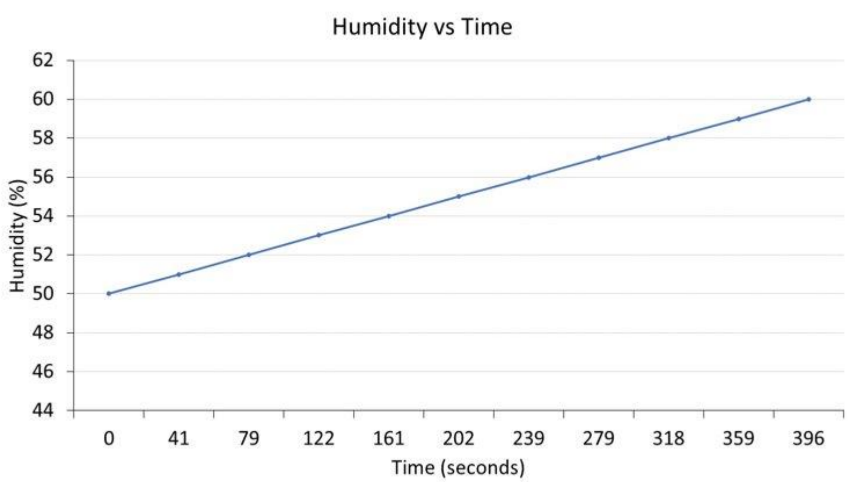


Figure 46: Humidity Vs Time Graph for Humidifier Being Turned On

TEMPERATURE VS TIME GRAPH FOR FAN BEING TURNED ON

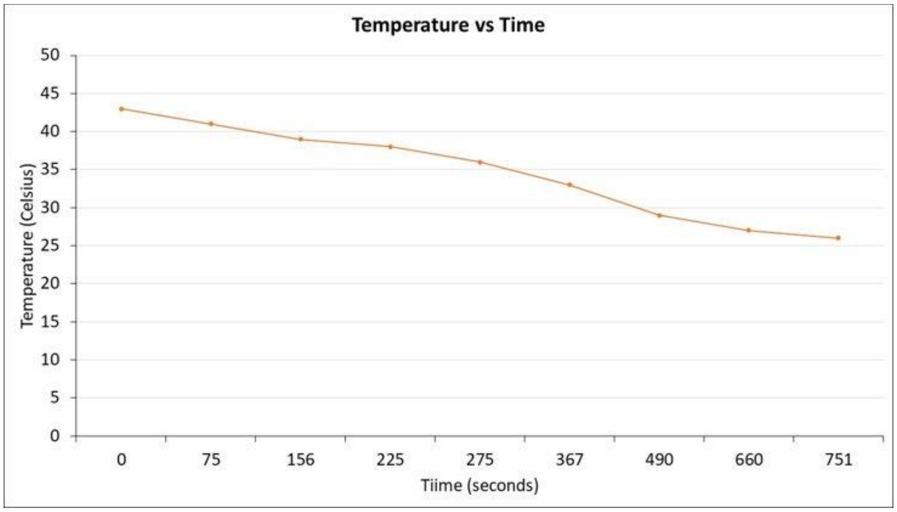


Figure 47: Temperature Vs Time Graph for Fan Being Turned On

EGG INCUBATOR AT THE OUTSIDE OF ROOM UNDER THE SUN

Figure 48: The egg incubator has been placed outside of the room which is under the sun.

TEMPERATURE VS TIME GRAPH FOR BULBS BEING TURNED ON

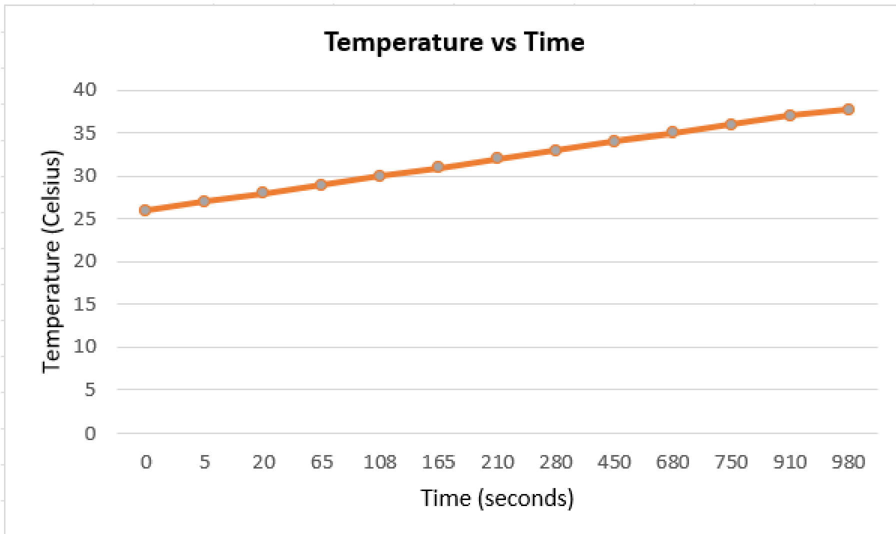


Figure 49: Temperature Vs Time Graph for Bulbs Being Turned On

HUMIDITY VS TIME GRAPH FOR BULBS BEING TURNED ON

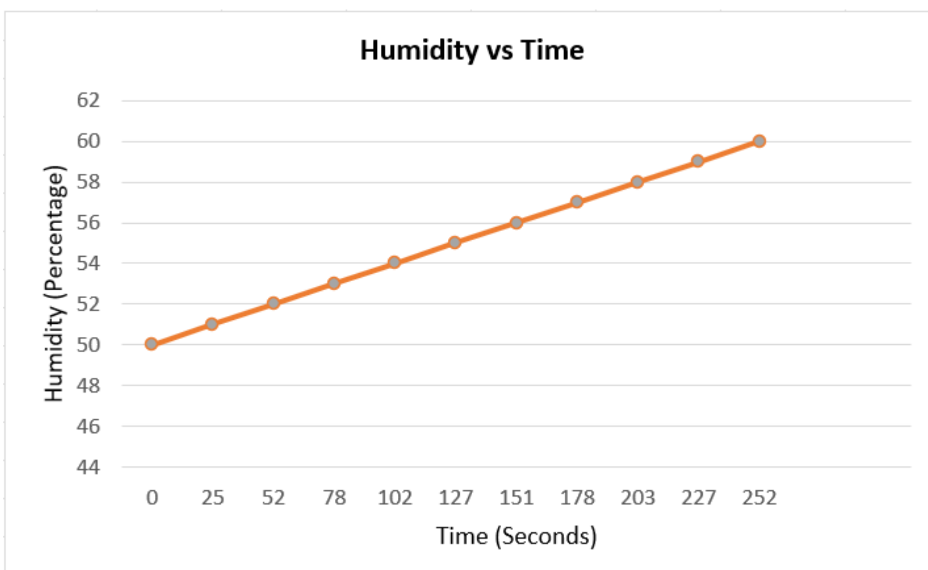


Figure 50: Humidity Vs Time Graph for Bulbs Being Turned On

TEMPERATURE VS TIME GRAPH FOR FAN BEING TURNED ON

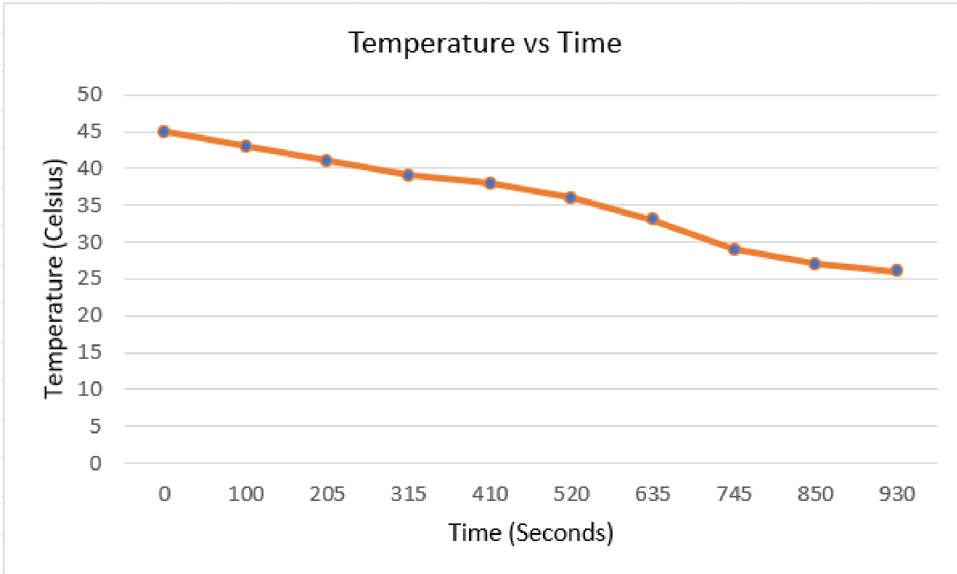


Figure 51: Temperature Vs Time Graph for Fan Being Turned On

8.4 DISCUSSION AND JUSTIFICATION FOR EGG INCUBATOR IN EACH ENVIRONMENT

8.4.1 DISCUSSION AND JUSTIFICATION OF TEMPERATURE VS TIME GRAPH WHEN BULB IS TURNED ON

Based on graph from 8.3.1.1, 8.3.2.1 and 8.3.3.1, we found out that the rate of temperature increase for the egg incubator being placed at the outside is the highest whereas the rate of temperature increase for the egg incubator being placed in room with air conditioning is the lowest along with the egg incubator being placed in a room with normal temperature at the middle.

Although it may seem that there are some high and low rate, but the time taken for the temperature to increase for each of these environments are not showing a large significant difference which means it is still within our acceptable range as the time taken for each of these environments are a few minutes different between each other.

8.4.2 DISCUSSION AND JUSTIFICATION OF HUMIDITY VS TIME GRAPH WHEN HUMIDIFIER IS TURNED ON

Based on graph from 8.3.1.2, 8.3.2.2 and 8.3.3.2, we found out that the rate of humidity increases for the egg incubator being placed at the outside is the highest whereas the rate of humidity increase for the egg incubator being placed in room with air conditioning is the lowest along with the egg incubator being placed in a room with normal temperature at the middle.
The time taken for the humidity to increase for each of these environments is not showing a large significant difference too which means it is still within our acceptable range due to its few minutes difference only.

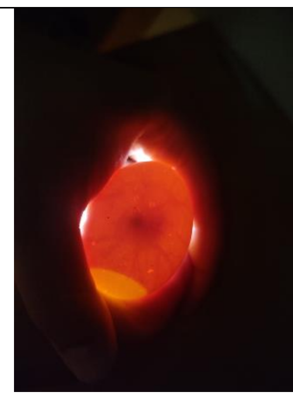
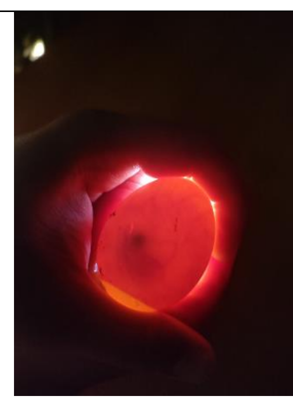
8.4.3 DISCUSSION AND JUSTIFICATION OF TEMPERATURE VS TIME GRAPH WHEN FAN IS TURNED ON

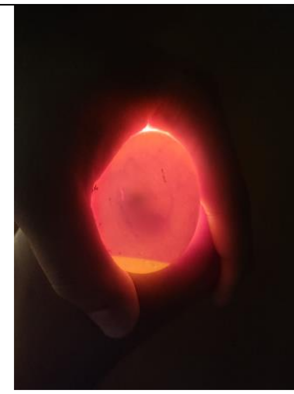
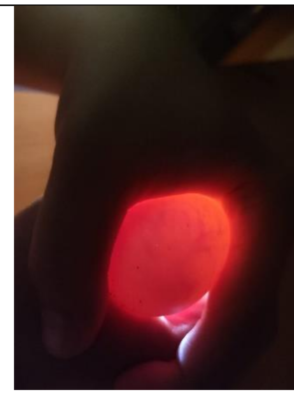
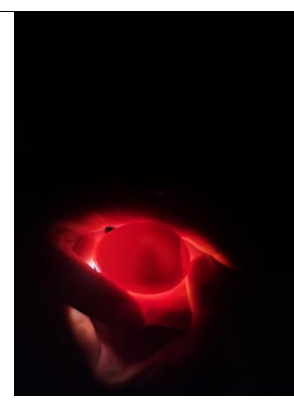
Based on graph from 8.3.1.3, 8.3.2.3 and 8.3.3.3, we found out that the rate of temperature decreases for the egg incubator being placed at the outside is the lowest whereas the rate of temperature decrease for the egg incubator being placed in room with air conditioning is the highest along with the egg incubator being placed in a room with normal temperature at the middle. The time taken for the temperature to drop for each of these environments does not show a large significant difference due to a few minutes different only between each other.

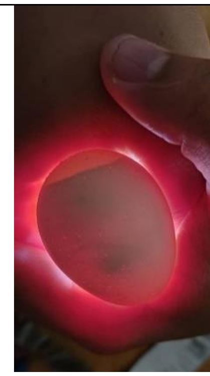
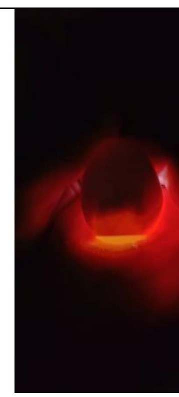
8.5 RESULT REPRESENTATION

Although we are testing our egg incubator in different kinds of environments, but the eggs were being incubate along the way so at the end we still achieved the same outcomes as the eggs has been changing and progressing differently day by day showing the eggs are in a good condition due to the good performance of our machine.

Days	Google References	Result
1		
2		
3		

4		
5		
6		
7		

8	 Day 8	
9	 Day 9	
10	 Day 10	
11	 Day 11	

12	 Day 12	
13	 Day 13	
14	 Day 14	
15	 Day 15	

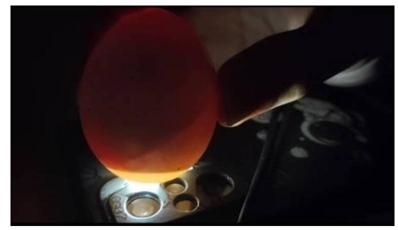
16		
17		
18		
19		
20		

Table 28: Result Representation

8.6 IMPROVEMENT PROPOSAL

Although the machine functions satisfied all our objectives, there are quite a few improvements that we need to make to have better quality of the product since it is just a prototype. First, would be the insulator inside the egg incubator, which this machine can still be improved by investigating material selection, surface treatment, thickness, design optimization and thermal management.

Secondly, the software, the data collected and sent from the egg incubator to ESP8266 to our phone, is still not stable, as we might experience some wrong data transferring. Although this has been solved by modifying the codes for the machine to restart or refresh, we believe there will be other ways for us to solve this problem far better than the restart or refresh function.

Third would be the electrical part, the cable management and the storing of the electric components in the machine are still not in a better position as we might experience some short circuit of the board like the relay causing the relay to be broken. This issue has been solved by separating any electric conducting material away from the electric components and the cables or wires has been labeled and arranged well in a junction box, but we believe that there might be a better to manage all the components and wires like by using cable management tools and electric separator tools.

CHAPTER 9

PRODCT OPERATION AND SAFETY

9.1 OPERATION MANUAL SOP

Standard Operating Procedure for Egg Incubator	
Department/Unit	All department and users
Purpose	The purpose of this SOP is to provide step-by-step on how to operate the Egg Incubator to avoid any accident on the user and machine.
Who can perform this SOP (STANDARD OF PROCEDURES)?	All department and users
Equipment/Tools Needed	
Personal Protective Equipment (PPE)	None
Tools	None
Reference Material	None
Others	None
Procedure	1.Turn on the main switch. a) Make sure the user's hands are not wet. b) Switch on the plug from the socket. c) Switch on the switch on the machine. 2. Putting eggs on the roller. a) After the machine has been turned on, leave the machine for around 15 minutes to pre heat until the desired temperature showing on the LCD. b) After the machine has been pre heated, place the eggs on the egg tray.

	c) Ensure there is not any electrical device like phone or watch left it inside as the heat might cause some damage to the devices. 3. Turn off the incubator when all eggs have already hatched, or the egg incubator is not being used anymore.
Prepared by **Reviewed by** **Approved by**	*NIVYA* NIVYA A/P K. RAJAKUMARAN (MANUFACTURING ENGINEER) *SYAHRIL* MOHAMAD SYAHRIL SYAFIQ BIN MOHD NIZAM (ELECTRICAL ENGINEER) *SHENG* WONG KAI SHENG (PROJECT LEADER)

Table 29: SOP

9.2 PRODUCT SAFETY DATASHEET

COMPANY IDENTIFICATION	
PRODUCT USE	Convenience product
IDENTIFIED USERS	Customer use
MANUFACTURER'S NAME	FARMTECH SDN BHD
EMAIL ADDRESS	admin.info@farmtech.com
EMERGENCY TEL	07-12345678

Table 30: Company Identification

HAZARD IDENTIFICATION	
SKIN CONTACT	None
EYE CONTACT	None
FRACTURE	None
INHALATION	None
ELECTRICAL CONTACT	Turn on main power source with dry hands

Table 31: Hazard Identification

HANDLING STORAGE	
HANDLING PROCEDURES	Always refer the SOP before operating the machine
STORAGE REQUIREMENT	None

Table 32: Handling Storage

CHAPTER 10

CONCLUSION AND RECOMMENDATIONS

In conclusion, the design book report on the egg incubator emphasizes the significance of this technology in handling hatching eggs after harvest. The paper goes into depth about the several difficulties the sector must deal with as well as the chances to enhance the present hatching procedure. The creation of the prototype, with its cutting-edge features and technologically sophisticated design, has the potential to revolutionize the egg hatching process and enhance the product's quality and marketability. The egg business and the larger agriculture sector may be significantly impacted if this equipment is successfully developed.

To increase the accuracy and dependability of the egg incubator, it is recommended that more study be done based on the report's conclusions. To improve the performance of the machine, it should be investigated whether sophisticated sensor technologies, artificial intelligence, and machine learning algorithms may be used. To make the machine more accessible and user-friendly, efforts should also be made to enhance the user experience. The egg incubator can significantly impact the industry and enhance the conditions for egg hatching with the correct investment in research and development.

REFERENCES

1. Purinamills.com. “21-Day Guide to Hatching Eggs: Purina Animal Nutrition.” Purinamills.com, 30 Nov. 2022, https://www.purinamills.com/chicken-feed/education/detail/hatching-eggs-at-home-a- 21-day-guide-for-baby-chicks#:~:text=Place%20the%20eggs%20in%20the,with%2050%2D55%20percent%20humidity.

2. Steve-chapple. “Egg Turning during Incubation - a Review by Mohamed Mostafa El-Ashram.” EmTech Hatchery Systems Ltd, 21 Aug. 2020, https://www.emtech-systems.com/technical-talk/egg-turning-during-incubation-a-revi ew-by-mohamed-mostafa-el-ashram/#:~:text=It%20is%20important%20to%20ensure,stopping%20will%20not%20affect%20hatchability.

3. “Egg Incubator DC Power 12V.” Get A Wholesale Egg Incubator Dc Power 12v For Your Poultry Business – Alibaba.com, https://www.alibaba.com/showroom/egg-incubator-dc-power-12v.html.

4. French NA (1997) Modeling Incubator Temperature: The Effect of Incubator Design, Embryonic Development, and Egg Size Poultry Sci, ,76: 124-133.

5. Mondal S & Pal S (2021) Low Cost Incubator Create New Income for Rural Villages, International Journal of Engineering Research & Technology (IJERT), Volume 9, Issue 11.PP:3-6.

6. A. Van Brecht (2005) ENVIRONMENT, WELL-BEING, AND BEHAVIOR Quantification of the Heat Exchange of Chicken Eggs, 353-361.

7. Bharosh K., Nischal P. (2021) Design, Fabrication, and Performance Analysis of an Automatic

Horizontal Egg Incubator, 77-85

8. Briscoe N D A and Wolfgang N(2018) Design of a Semi-Automatic Artificial

Incubator, European Journal of Applied Engineering and Scientific Research,6(3);4-

14

9. doko E , Ogbeh GO, Ikule FT,(2019), Design and Implementation of Automatic

Fixed Factors Egg Incubator, International Journal for Innovative Research in

Multidisciplinary Field, , Volume - 5, Issue - 6, June –,PP:1-8

10. https://alec.unl.edu/documents/cde/2017/livestock-management/incubating-and-hatching-eggs-2017.pdf

APPENDICES

A: Mechanical Drawings

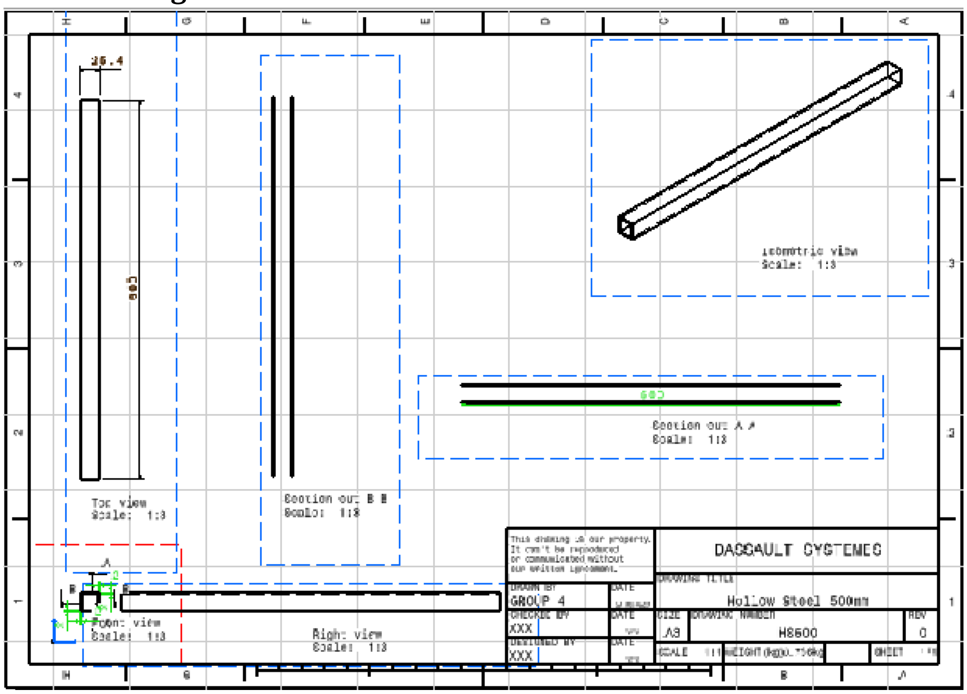


Figure 52: Hollow Steel 500mm

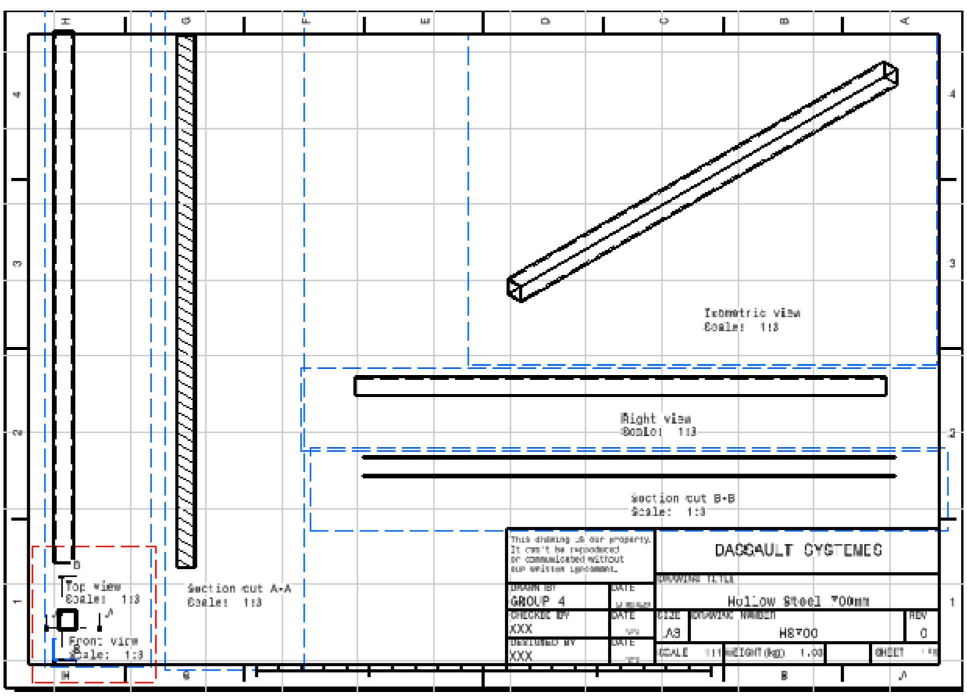


Figure 53: Hollow Steel 700mm

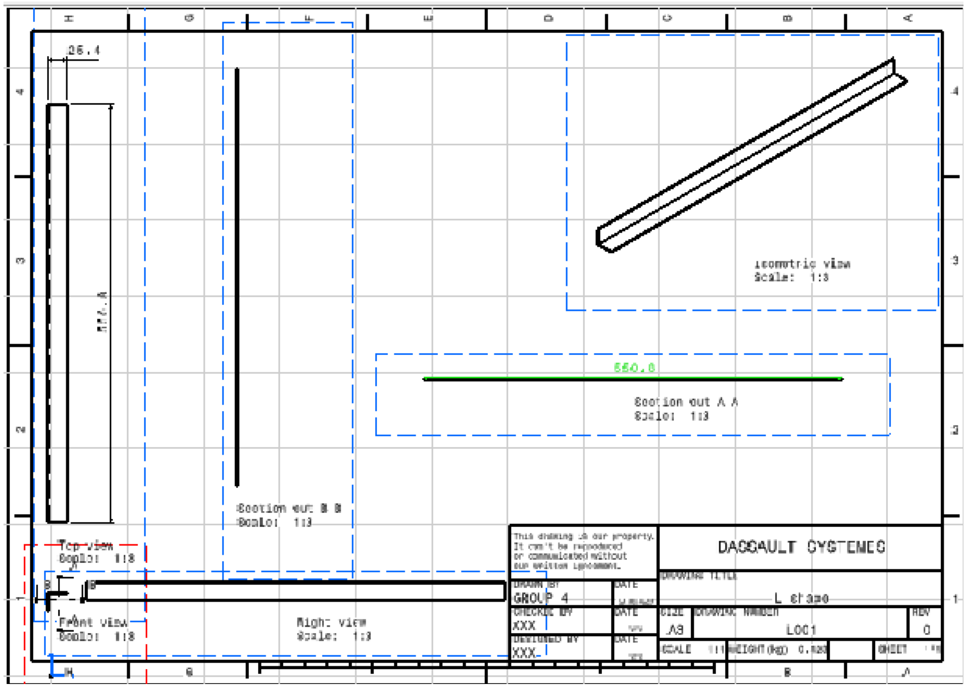


Figure 54: L shape

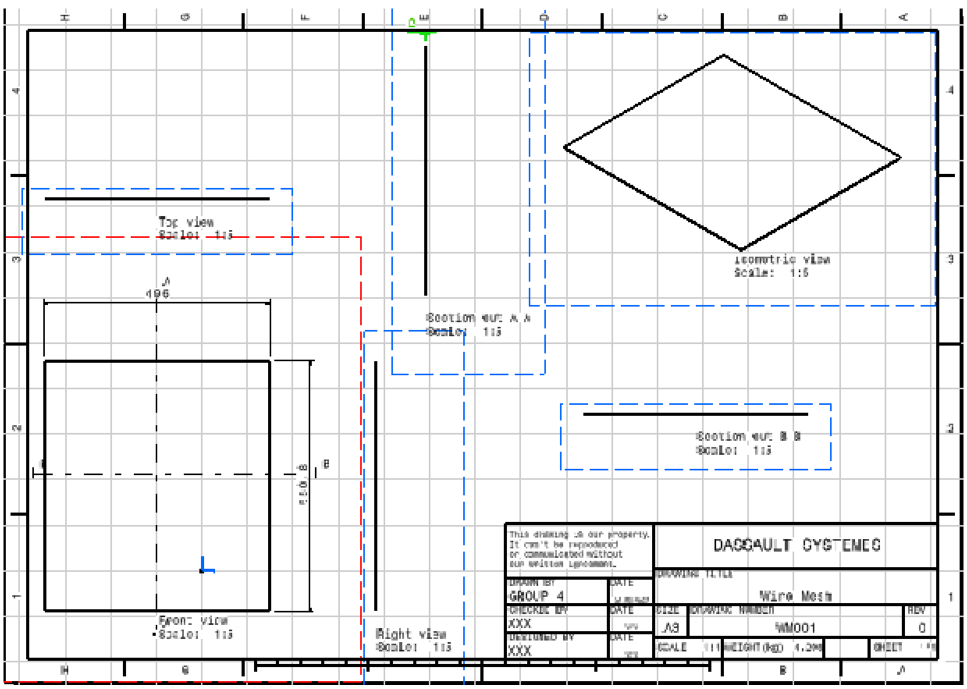


Figure 55: Wire Mesh

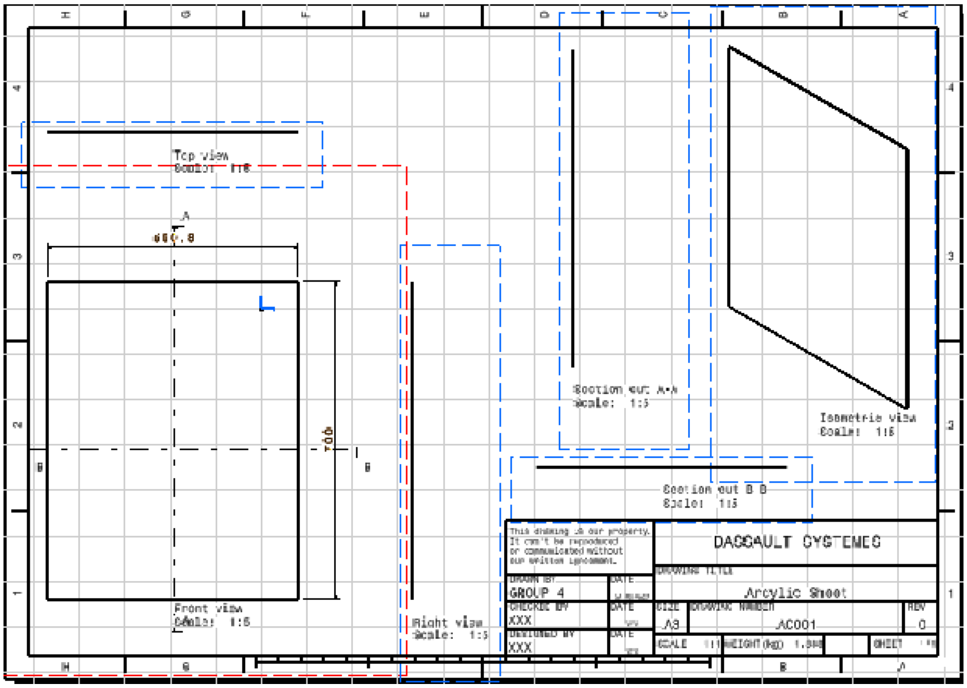


Figure 56: Acrylic Sheet

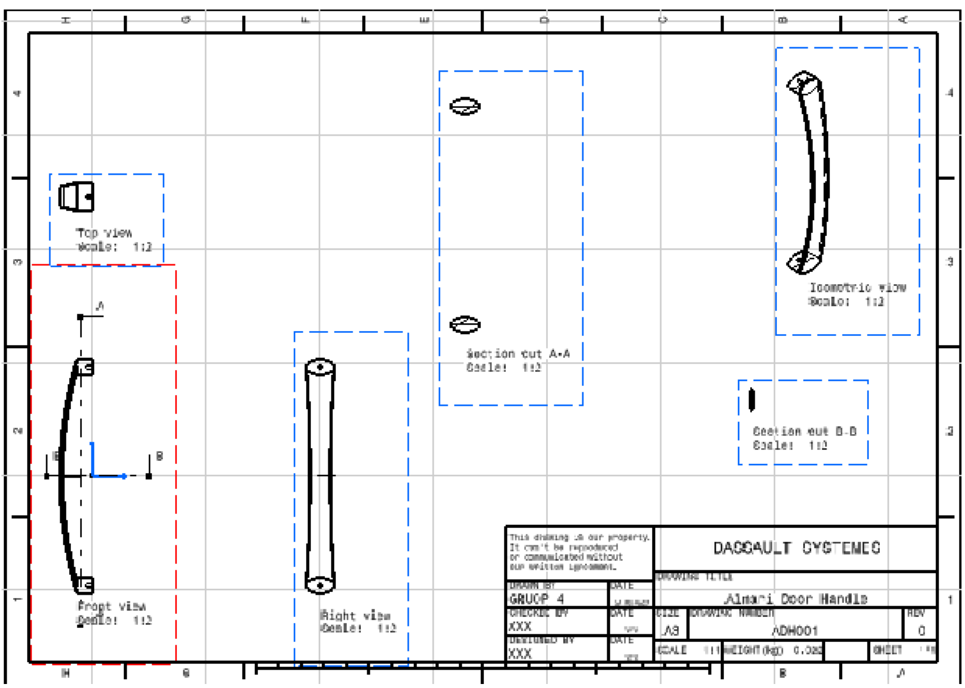


Figure 57: Door Handle

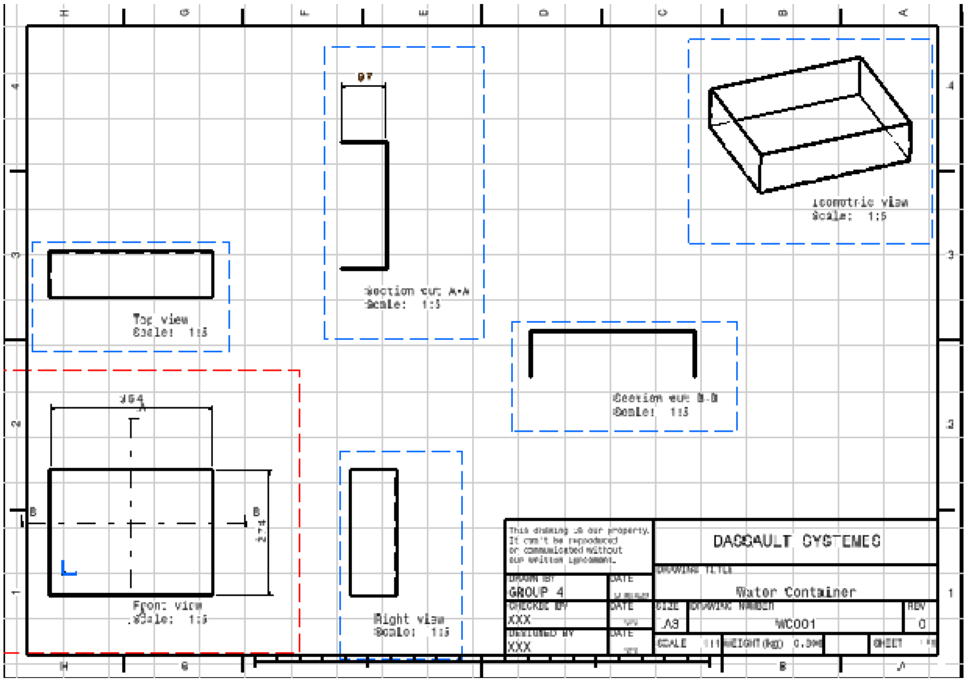


Figure 58: Water Container

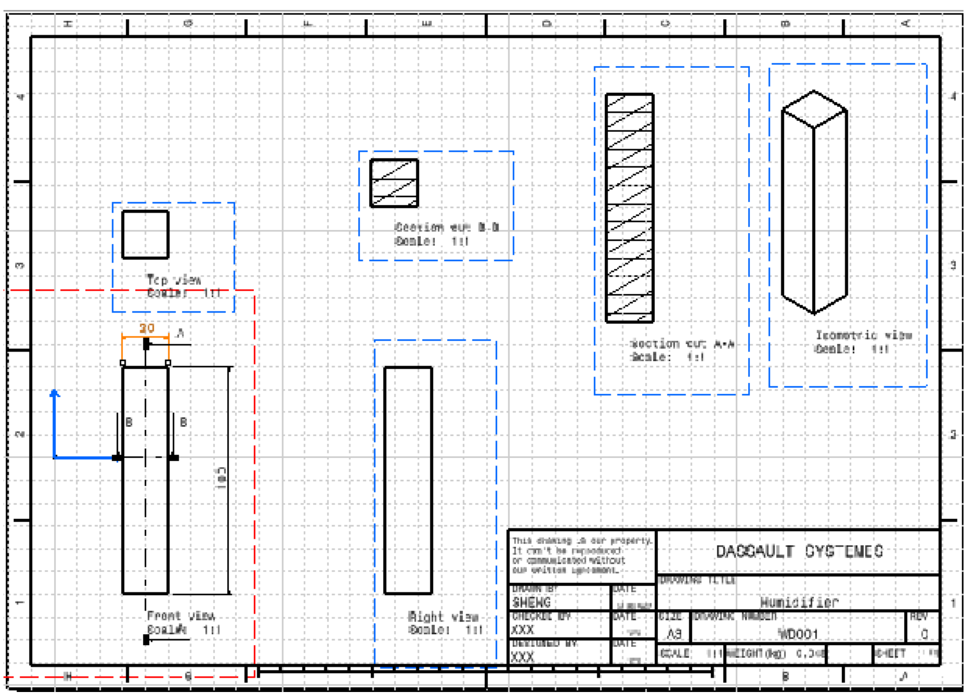


Figure 59: Humidifier

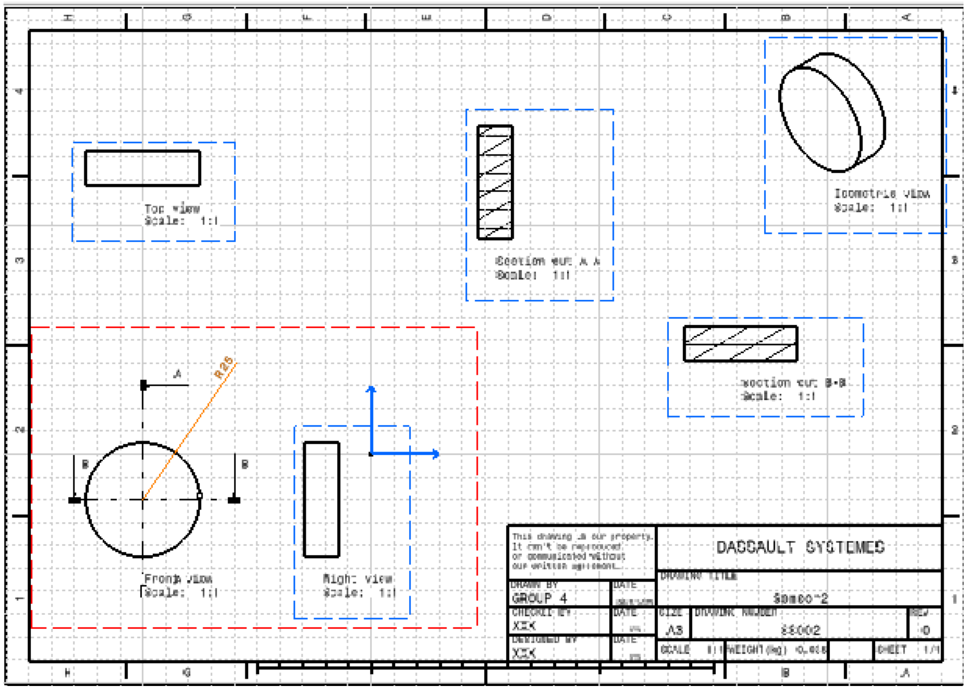


Figure 60: Sensor

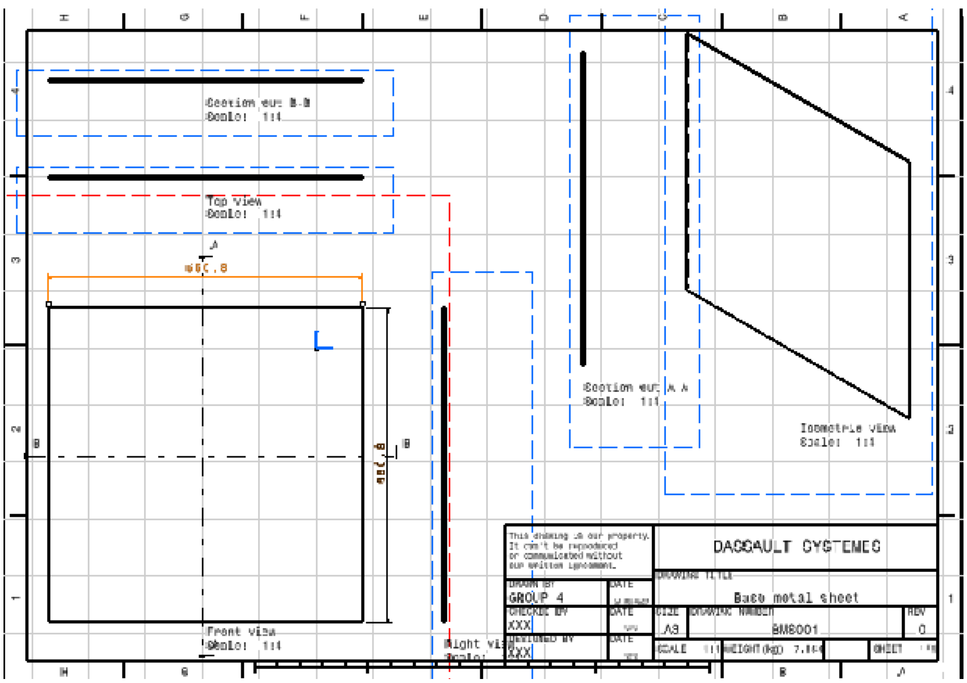


Figure 61: Base Metal Sheet

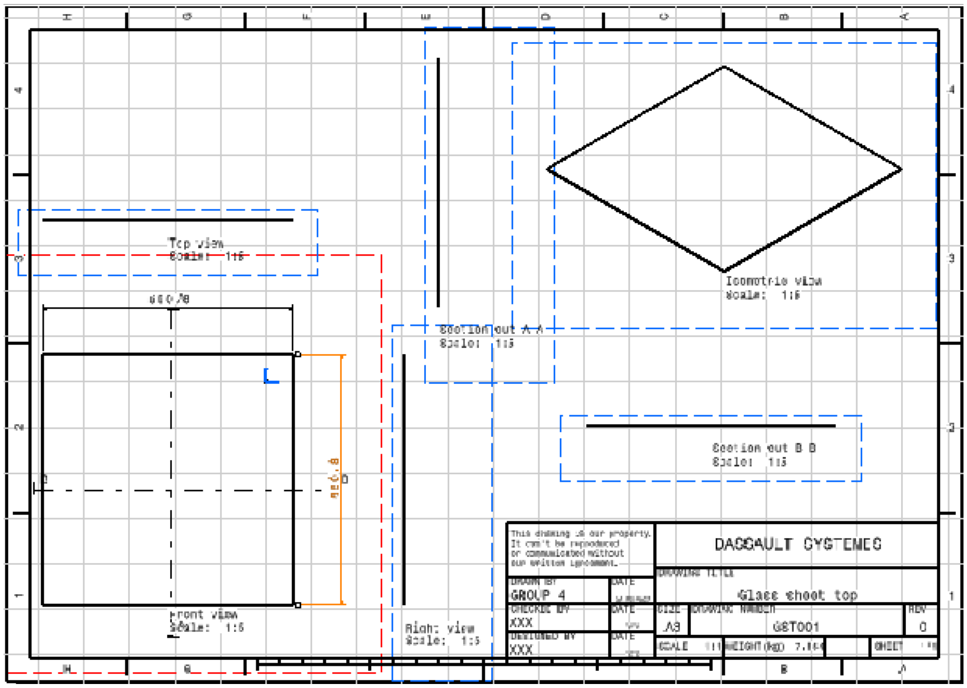


Figure 62: Acrylic Sheet Top

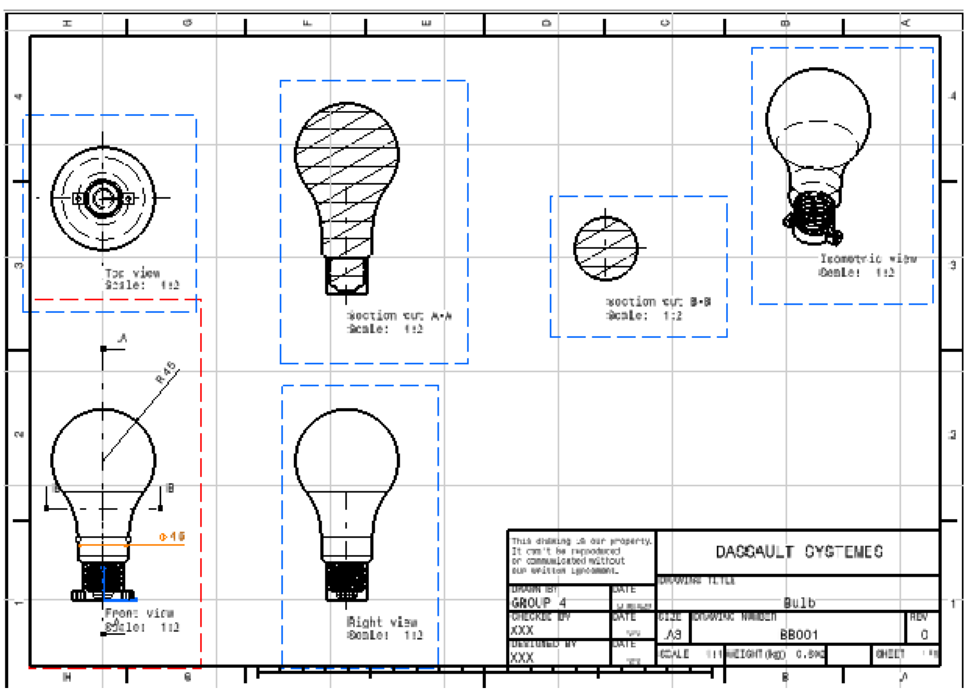


Figure 63: Bulb

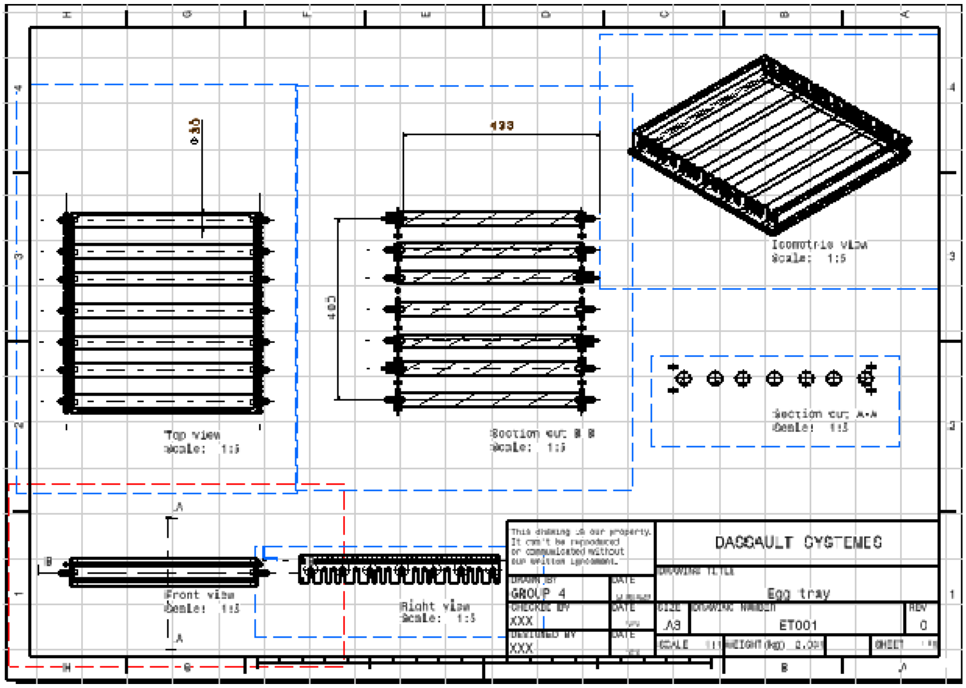


Figure 64: Egg tray

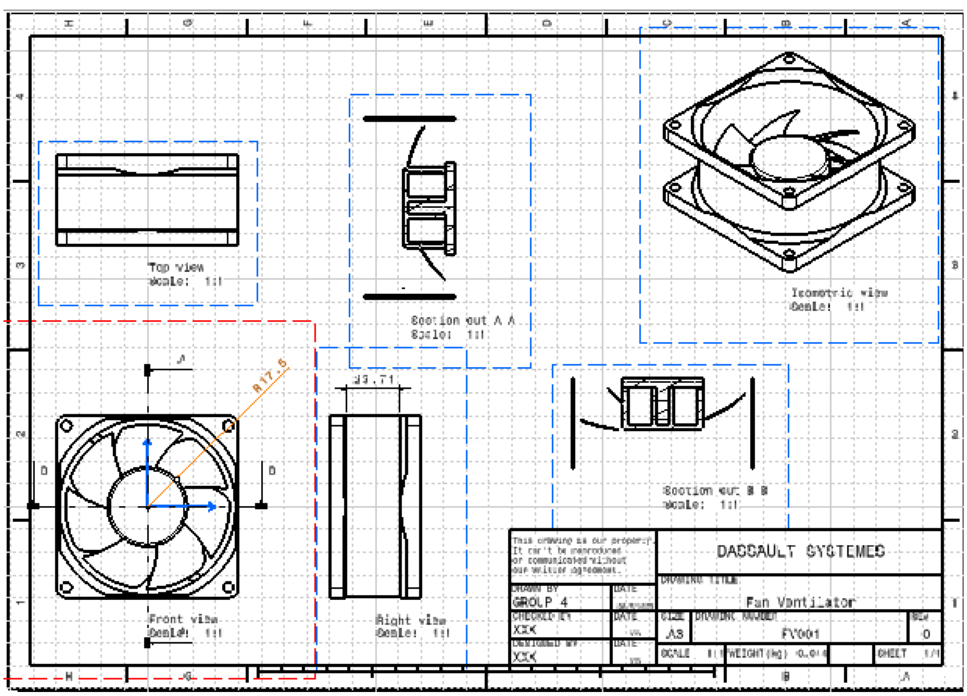


Figure 65: Fan Ventilator

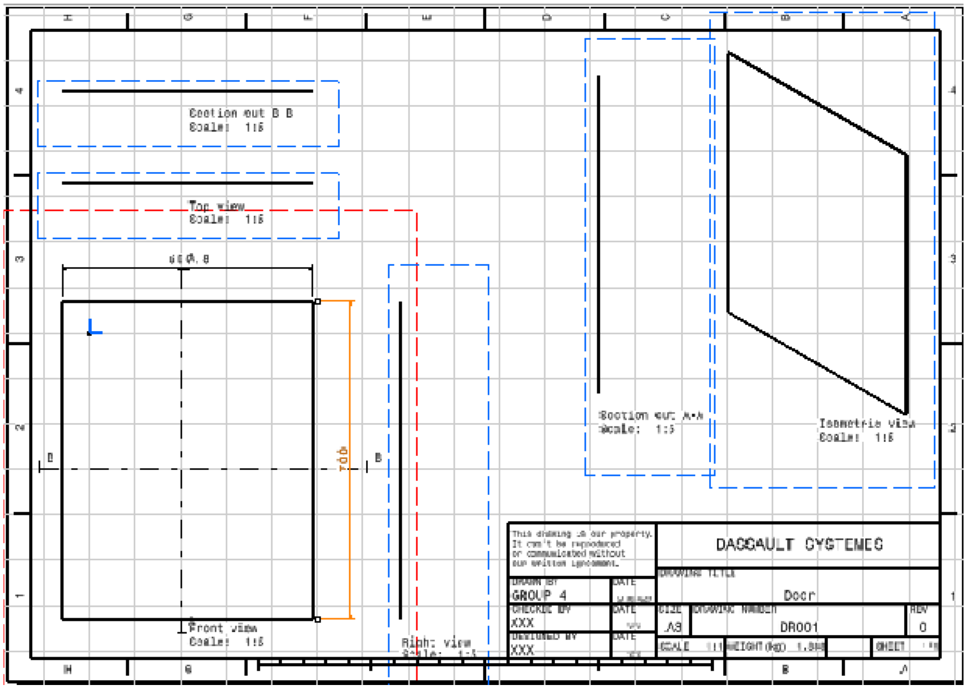


Figure 66: Door

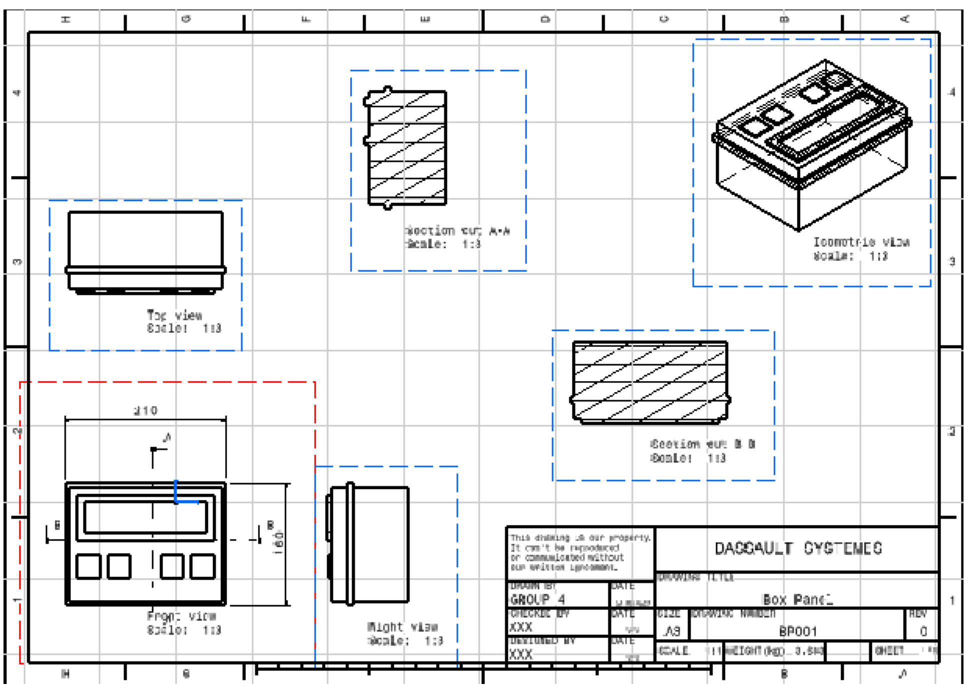


Figure 67: Box Panel

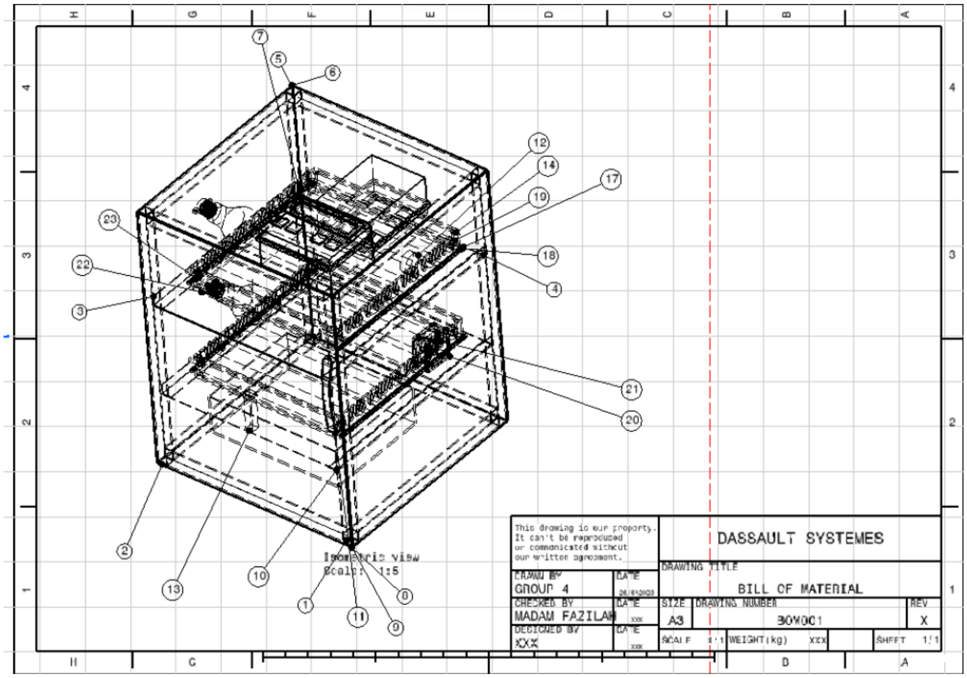


Figure 68: Bill of material

Bill of Material: Product1

Number	Name	Part Number	Definition	Quantity	Product Description	Source	Revision
1	Hollow Steel 500mm.1	HS500	Steel	8	Hollow Steel 500mm	Made	0
2	Hollow Steel 700mm.1	HS700	Steel	4	Hollow Steel 700mm	Made	0
3	L shape.1	L001	Steel	4	L shape	Made	0
4	wire mesh.1	WM001	Steel	2	Wire Mesh	Bought	0
	Product1.1.1	ET001	Plastic	2	Egg tray	Bought	0
5	Acrylic sheet.1	AC001	Plexiglass	3	Acrylic sheet	Made	0
6	glass sheet top.1	GST001	Steel	1	glass sheet top	Made	0
7	box panel.1	BP001	Plastic	2	box panel	Made	0
8	Base metal sheet.1	BMS001	Steel	1	Base metal sheet	Made	0
	FV001.1	FV001	Plastic	1	Fan Ventilator	Bought	0
	Product2.1	BB001	Plastic	2	Bulb	Bought	0
9	door .1	DR001	Plexiglass	1	Door	Made	0
10	water container.1	WC001	Plastic	1	water container	Bought	0
11	ALMARI_DOOR_HANDLE.1	ADH001	Plastic	1	ALMARI_DOOR_HANDLE	Bought	0
12	sensor 2.1	SS002	Plastic	1	sensor 2	Bought	0
13	water diffuser.1	WD001	Plastic	1	water diffuser	Bought	0

Bill of Material: ET001

Number	Name	Part Number	Definition	Quantity	Product Description	Source	Revision
14	Part1.1	Part1.2		1		Unknown	
15	Part2.1	Part2.2		1		Unknown	
16	Part3.1	Part3		1		Unknown	
	Product2.1	Product2		7		Unknown	
17	Part6.1	Part6		2		Unknown	

Bill of Material: Product2

Number	Name	Part Number	Definition	Quantity	Product Description	Source	Revision
18	Part5.1	Part5		2		Unknown	
19	Part4.1	Part4		1		Unknown	

Bill of Material: FV001

Number	Name	Part Number	Definition	Quantity	Product Description	Source	Revision
20	FH001.1	FH001	Fan holder	1		Bought	
21	F002.1	F002	Fan	1		Bought	

Bill of Material: BB001

Number	Name	Part Number	Definition	Quantity	Product Description	Source	Revision
22	Bulb Holder.1	Bulb Holder		1		Unknown	
23	Bulb.1	Bulb		1		Unknown	
24	Screw.1	Screw		1		Unknown	
25	Screw.2	Screw-2		1		Unknown	

Figure 69: Bill of Materials

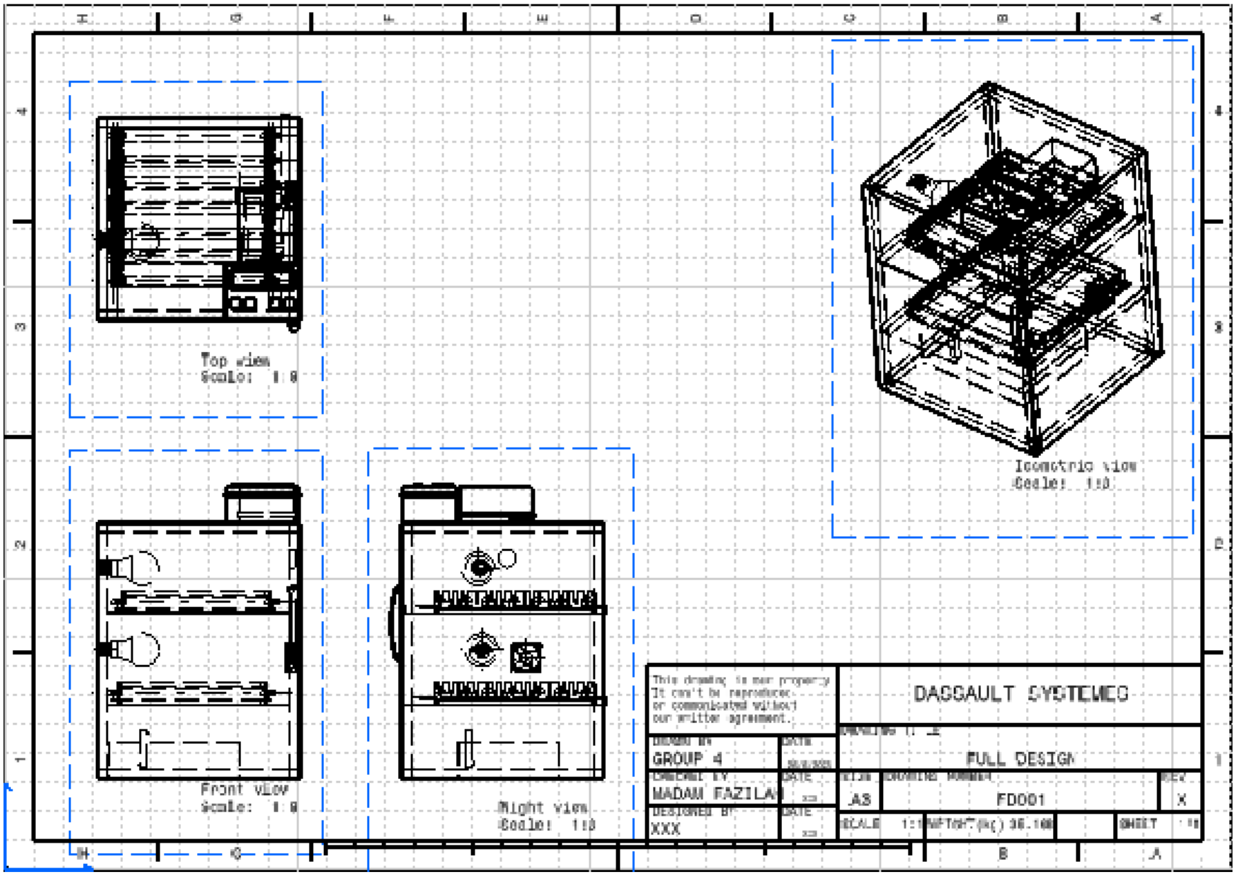


Figure 70: Full Design of egg incubator

B: Electrical Circuit Drawings

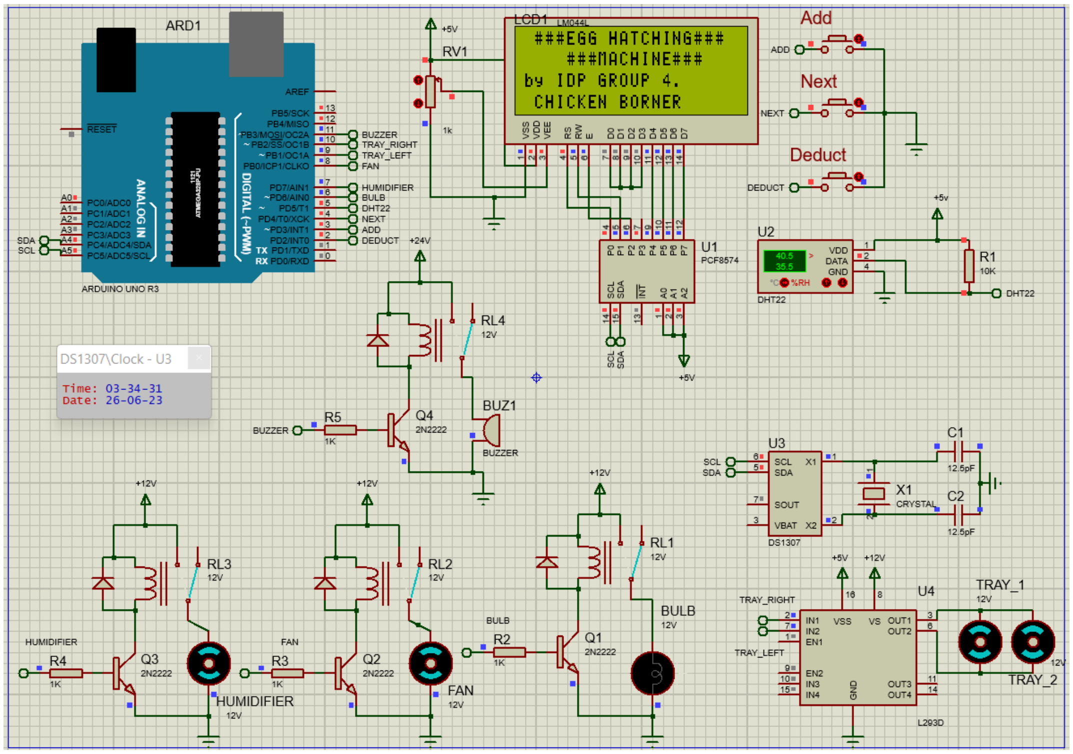


Figure 71: Electrical Circuit

C: Arduino Uno Software Codes

Due to the codes is too long, it has been uploaded to google drive in the link below:
https://drive.google.com/drive/folders/1Koq8gFaozONAf8DKZPLm2jem8ofWfgHk?usp=sharing

```
/* LIBRARIES INCLUDED */
#include <Arduino.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>
#include <LiquidCrystal_I2C.h>
#include <EEPROM.h>
#include "RTClib.h"
#include <TimerOne.h>
#include <SoftwareSerial.h>
#include <avr/wdt.h>


/* Definition of variables */
#define PRESSED  0
#define NOTPRESSED 1
#define DEDUCT 2
#define ADD 3
#define NEXT 4
#define DHTPIN 5
#define DHTTYPE DHT22 // NEED TO CHANGE LATER //
#define BULB 6
#define HUMIDIFIER 7
#define FAN 8
#define TRAY_LEFT 9
#define TRAY_RIGHT 10
#define BUZZER 11
#define reset A0


// Serial Communication //
SoftwareSerial espSerial(12, 13);


/* Declaration of variable */
DateTime now;
DateTime end;
```

```
unsigned int     a = 2003;
unsigned short   b = 0;
unsigned short   c = 0;
unsigned short   d = 0;
unsigned short   e = 0;
unsigned short   f = 0;
unsigned short   n = 1;


unsigned long count;
unsigned long count1;


bool             motor_direction = 0;
bool             millis_flag = 1;
bool             Home = 1;
int              rotation_duration = 30;
int              first_upload = 5;
float            humidity;
float            humid_ref = 57.0;
float            humid_ref_tol = 2.0;
float            humid_ref_tres = 3;
float            temperature;
float            temp_ref = 37.7;
float            temp_ref_tol = 0.2;
float            temp_ref_tres = 0.3;
unsigned short   SPIN_time = 2;
unsigned short   SPIN_hour = 2;
unsigned short   SPIN_minute = 59;
unsigned short   SPIN_second = 59;
unsigned short   incub_days_total = 6;
unsigned short   incub_days_left = 6;


float temp_ref_holder;
float humid_ref_holder;
float temp_ref_tol_holder;
float humid_ref_tol_holder;
float temp_ref_tres_holder;
float humid_ref_tres_holder;
unsigned short   SPIN_time_holder;
unsigned short   SPIN_hour_holder = 2;
unsigned short   incub_days_left_holder;
```

```
int   incub_days_total_holder;
int   rotation_duration_holder;


short int   SPIN_hour_address = 55;
short const motor_dir_address = 33;
short const temp_ref_address = 0;
short const humid_ref_address = 10;
short const temp_ref_tol_address = 20;
short const humid_ref_tol_address = 30;
short const SPIN_time_address = 40;
short const incub_days_total_address = 50;
short const temp_ref_tres_address = 60;
short const humid_ref_tres_address = 70;
short const incub_days_left_address = 80;
short const startday_year_address = 81;
short const startday_month_address = 82;
short const startday_day_address = 83;
short const startday_hour_address = 84;
short const startday_minute_address = 85;
short const startday_second_address = 86;
int         rotation_duration_address = 100;
int         first_upload_address = 200;
String slogan = "CHICKEN BORNER";


RTC_DS1307 rtc;
LiquidCrystal_I2C lcd(0x27, 20, 4);
DHT dht(DHTPIN, DHTTYPE);


byte customChar[] = {
  B00100,
  B00100,
  B01110,
  B10001,
  B10001,
  B10001,
  B01110,
  B00100
};


byte customChar1[] = {
  B00100,
```

```
  B00100,
  B01110,
  B11111,
  B11111,
  B11111,
  B01110,
  B00100
};


byte arrow[8] = {
  0b01000,
  0b01100,
  0b01110,
  0b01111,
  0b01110,
  0b01100,
  0b01000,
  0b00000
};


void  ft_initialize(void)
{
  digitalWrite(TRAY_LEFT, LOW);
  digitalWrite(TRAY_RIGHT, LOW);


  rtc.begin();
  now = rtc.now();
  dht.begin();
  lcd.init();
  lcd.backlight();


  lcd.setCursor(1, 0);
  lcd.print("###EGG HATCHING###");
  lcd.setCursor(4, 1);
  lcd.print("###MACHINE###");
  lcd.setCursor(0, 2);
  lcd.print("by IDP GROUP 4.");
  delay(500);
  lcd.setCursor(1, 3);
  for (int i = 0; i < 17; i++)
  {
```

```
  lcd.setCursor(i + 1, 3);
  lcd.print(slogan[i]);
  delay(100);
}
delay(20);
first_upload = EEPROM.read(first_upload_address);
delay(20);


if (first_upload == 255)
{
  EEPROM.write(first_upload_address, 10);
  delay(20);
  EEPROM.put(temp_ref_address, temp_ref);
  delay(20);
  EEPROM.put(temp_ref_tol_address, temp_ref_tol);
  delay(20);
  EEPROM.put(humid_ref_address, humid_ref);
  delay(20);
  EEPROM.put(humid_ref_tol_address, humid_ref_tol_address);
  delay(20);
  EEPROM.put(temp_ref_tres_address, temp_ref_tres);
  delay(20);
  EEPROM.put(humid_ref_tres_address, humid_ref_tres);
  delay(20);
  EEPROM.write(incub_days_total_address, incub_days_total);
  delay(20);
  EEPROM.write(SPIN_time_address, SPIN_time);
  delay(20);
  EEPROM.write(SPIN_hour_address, SPIN_hour);
  delay(20);
  EEPROM.write(incub_days_left_address, incub_days_left);
  delay(20);
  EEPROM.write(SPIN_time_address, 2);
  delay(20);
  EEPROM.write(rotation_duration_address, rotation_duration);
}
delay(20);
EEPROM.get(temp_ref_address, temp_ref_address);
delay(20);
EEPROM.get(temp_ref_tol_address, temp_ref_tol_address);
delay(20);
EEPROM.get(humid_ref_address, humid_ref_address);
delay(20);
EEPROM.get(humid_ref_tol_address, humid_ref_tol_address);
```

```
  delay(20);
  EEPROM.get(temp_ref_tres_address, temp_ref_tres_address);
  delay(20);
  EEPROM.get(humid_ref_tres_address, humid_ref_tres_address);


  incub_days_total = EEPROM.read(incub_days_total_address);
  delay(20);
  SPIN_time = EEPROM.read(SPIN_time_address);
  delay(20);
  SPIN_hour = EEPROM.read(SPIN_hour_address);
  delay(20);
  rotation_duration = EEPROM.read(rotation_duration_address);
  delay(20);
  incub_days_left = EEPROM.read(incub_days_left_address);
  delay(20);
  lcd.clear();
}


void  ft_LED_blink(void)
{
  if (SPIN_second > 0)
    SPIN_second--;
}


void ft_reset()
{
  wdt_enable(WDTO_15MS);
  while(1);
}


void  setup()
{
  digitalWrite(reset, HIGH);
  pinMode(reset, OUTPUT);
  pinMode(NEXT, INPUT_PULLUP);
  pinMode(ADD,INPUT_PULLUP);
  pinMode(DEDUCT,INPUT_PULLUP);
  pinMode(FAN,OUTPUT);
  pinMode(BULB,OUTPUT);
  pinMode(BUZZER,OUTPUT);
  pinMode(TRAY_LEFT,OUTPUT);
```

```
  pinMode(TRAY_RIGHT,OUTPUT);
  pinMode(HUMIDIFIER,OUTPUT);
  ft_initialize();
  lcd.createChar(0,customChar);
  lcd.createChar(1,customChar1);
  lcd.createChar(10,arrow);


  a = EEPROM.read(startday_year_address) + 1792;
  delay(20);
  b = EEPROM.read(startday_month_address);
  delay(20);
  c = EEPROM.read(startday_day_address);
  delay(20);
  d = EEPROM.read(startday_hour_address);
  delay(20);
  e= EEPROM.read(startday_minute_address);
  delay(20);
  f= EEPROM.read(startday_second_address);
  delay(20);


  delay(20);
  DateTime startday(2021,7,5,2,48,0);
  Timer1.initialize(1000000);
  Timer1.attachInterrupt(ft_LED_blink); // blinkLED to run every 0.15 seconds
  Serial.begin(115200);
  espSerial.begin(9600); // Serial Communication //
}


void  ft_read_temphumid(void)
{
  temperature = dht.readTemperature();
  humidity = dht.readHumidity();
  lcd.setCursor(6, 0);
  lcd.print(temperature);
  lcd.setCursor(7, 1);
  lcd.print(humidity);
}


void  ft_buzzer()
{
  digitalWrite(BUZZER, HIGH);
```

```
  delay(100);
  digitalWrite(BUZZER, LOW);
  delay(40);
}


void  ft_save_param(void)
{
  delay(20);
  EEPROM.put(temp_ref_address, temp_ref);
  delay(20);
  EEPROM.put(temp_ref_tol_address, temp_ref_tol);
  delay(20);
  EEPROM.put(humid_ref_address, humid_ref);
  delay(20);
  EEPROM.put(humid_ref_tol_address, humid_ref_tol_address); //here changes motor direction
  delay(20);
  EEPROM.put(temp_ref_tres_address, temp_ref_tres);
  delay(20);
  EEPROM.put(humid_ref_tres_address, humid_ref_tres);
  delay(20);
  EEPROM.put(incub_days_total_address, incub_days_total);
  delay(20);
  incub_days_left = incub_days_total;
  delay(20);
  EEPROM.write(incub_days_left_address,incub_days_left);
  delay(20);


  DateTime startday(a,b,c,d,e,f);
  startday = rtc.now();


  EEPROM.write(startday_year_address,startday.year());
  delay(20);
  EEPROM.write(startday_month_address,startday.month());
  delay(20);
  EEPROM.write(startday_day_address,startday.day());
  delay(20);
  EEPROM.write(startday_hour_address,startday.hour());
  delay(20);
  EEPROM.write(startday_minute_address,startday.minute());
  delay(20);
  EEPROM.write(startday_second_address,startday.second());
  delay(20);
```

```
  EEPROM.write(SPIN_time_address, SPIN_time);
  delay(20);
  EEPROM.write(rotation_duration_address, rotation_duration);
  delay(20);
  SPIN_hour = SPIN_time - 1;
  delay(20);
  EEPROM.write(SPIN_hour_address, SPIN_hour);
  delay(20);


  a = EEPROM.read(startday_year_address) + 1792;
  delay(20);
  b = EEPROM.read(startday_month_address) ;
  delay(20);
  c = EEPROM.read(startday_day_address);
  delay(20);
  d = EEPROM.read(startday_hour_address);
  delay(20);
  e= EEPROM.read(startday_minute_address);
  delay(20);
  f= EEPROM.read(startday_second_address);
  delay(20);
  rotation_duration_address = rotation_duration;


  if(SPIN_time)
  {
    SPIN_second = 59;
  }
  SPIN_time = 2;
  SPIN_second = 5;
  SPIN_minute = 0;
  SPIN_hour = 0;
}


void  ft_temp_regulation(void)
{
  if (temperature == temp_ref)
  {
    digitalWrite(FAN, LOW);
    digitalWrite(BULB, LOW);
  }
```

```
  else if ((temperature >= (temp_ref - temp_ref_tol)) && (temperature <= (temp_ref +
temp_ref_tol)))
  {
    digitalWrite(FAN, LOW);
    digitalWrite(BULB, LOW);
    lcd.setCursor(16, 0);
    lcd.write(0);
  }
  ft_read_temphumid();
  if (temperature > (temp_ref + temp_ref_tol))
  {
    digitalWrite(FAN, HIGH);
    digitalWrite(BULB, LOW);


    if (temperature >= (temp_ref + temp_ref_tol + temp_ref_tres))
    {
      // ft_buzzer();
      digitalWrite(BUZZER, HIGH);
      lcd.setCursor(16, 0);
      lcd.write(1);
      delay(100);
      digitalWrite(BUZZER, LOW);
      lcd.setCursor(16, 0);
      lcd.print(" ");
      delay(100);
    }
    lcd.setCursor(16, 0);
    lcd.write(1);
  }
  ft_read_temphumid();
  if (temperature < (temp_ref - temp_ref_tol - 0.1))
  {
    digitalWrite(FAN, LOW);
    digitalWrite(BULB, HIGH);


    if (temperature <= (temp_ref - temp_ref_tol - temp_ref_tres - 0.1))
    {
      ft_buzzer();
      lcd.setCursor(16, 0);
      lcd.write(1);
      delay(100);
      lcd.setCursor(16, 0);
      lcd.print(" ");
```

```
      delay(100);
    }
    lcd.setCursor(16, 0);
    lcd.write(1);
  }
  if (digitalRead(BULB) == HIGH)
  {
    espSerial.print("%"); // Serial Communication //
    espSerial.print(1); // Serial Communication //
    delay(200);
  }
  else
  {
    espSerial.print("%"); // Serial Communication //
    espSerial.print(0); // Serial Communication //
    delay(200);
  }
  if (digitalRead(FAN) == HIGH)
  {
    espSerial.print("^"); // Serial Communication //
    espSerial.print(1); // Serial Communication //
    delay(200);
  }
  else
  {
    espSerial.print("^"); // Serial Communication //
    espSerial.print(0); // Serial Communication //
    delay(200);
  }
}


void  ft_humid_regulation()
{
  if ((humidity >= (humid_ref - humid_ref_tol)) && (humidity <= (humid_ref + humid_ref_tol)))
  {
    digitalWrite(HUMIDIFIER, LOW);
    // digitalWrite(FAN, LOW);
    lcd.setCursor(16, 1);
    lcd.write(0);
  }
  ft_read_temphumid();
  if (humidity > (humid_ref + humid_ref_tol + 0.1))
  {
```

```
    digitalWrite(HUMIDIFIER, LOW);
    // digitalWrite(FAN, HIGH);
    // delay(10000);
    if (humidity >= (humid_ref + humid_ref_tol + humid_ref_tres + 0.1))
    {
      ft_buzzer();
      lcd.setCursor(16, 1);
      lcd.write(1);
      delay(100);
      lcd.setCursor(16, 1);
      lcd.print(" ");
      delay(100);
    }
    lcd.setCursor(16, 1);
    lcd.write(1);
  }
  ft_read_temphumid();
  if (humidity < (humid_ref - humid_ref_tol))
  {
    digitalWrite(HUMIDIFIER, HIGH);
    // digitalWrite(FAN, LOW);
    if (humidity <= (humid_ref - humid_ref_tol - humid_ref_tres - 0.1))
    {
      ft_buzzer();
      lcd.setCursor(16, 1);
      lcd.write(1);
      delay(100);
      lcd.setCursor(16, 1);
      lcd.print(" ");
      delay(100);
    }
    lcd.setCursor(16,1);
    lcd.write(1);
  }
  if (digitalRead(HUMIDIFIER) == HIGH)
  {
    espSerial.print("&"); // Serial Communication //
    espSerial.print(1); // Serial Communication //
    delay(200);
  }
  else
  {
    espSerial.print("&"); // Serial Communication //
    espSerial.print(0); // Serial Communication //
    delay(200);
```

```
  }
}


void milsecdelay(){


    if (millis_flag){
      millis_flag=0;
      count =millis();
    }


    count1=millis();
    if ((count1-count)>=50)
    {
      // action
      millis_flag=1;
    }
}


void  ft_discard_param()
{
  temp_ref = temp_ref_holder;
  humid_ref = humid_ref_holder;
  temp_ref_tol = temp_ref_tol_holder;
  humid_ref_tol = humid_ref_tol_holder;
  temp_ref_tres = temp_ref_tres_holder;
  humid_ref_tres = humid_ref_tres_holder;
  SPIN_time = SPIN_time_holder;
  incub_days_total = incub_days_total_holder;
  incub_days_left = incub_days_left_holder;
  SPIN_hour = SPIN_hour_holder;
  rotation_duration = rotation_duration_holder;
}


void  ft_menu()
{
  lcd.clear();
  lcd.setCursor(3, 0);
  lcd.print("ENTERING MENU");
  lcd.setCursor(4, 1);
  lcd.print("HOLD ON...");
```

```
temp_ref_holder = temp_ref;
humid_ref_holder = humid_ref;
temp_ref_tol_holder = temp_ref_tol;
humid_ref_tol_holder = humid_ref_tol;
temp_ref_tres_holder = temp_ref_tres;
humid_ref_tres_holder = humid_ref_tres;
SPIN_time_holder = SPIN_time;
incub_days_total_holder = incub_days_total;
incub_days_left_holder = incub_days_left;
SPIN_hour_holder = SPIN_hour;
rotation_duration_holder = rotation_duration;


delay(200);
lcd.clear();


do
{
  lcd.setCursor(4, 1);
  lcd.print(">>");
  lcd.setCursor(1, 0);
  lcd.print("Set Temp: ");
  lcd.setCursor(6, 1);
  lcd.print(temp_ref);
  lcd.setCursor(10, 1);
  lcd.print((char) 223);
  lcd.setCursor(11, 1);
  lcd.print("C");


  lcd.setCursor(2, 2);
  lcd.print("Temp TOL Value:");
  lcd.setCursor(6, 3);
  lcd.print(temp_ref_tol);
  lcd.setCursor(10, 3);
  lcd.print((char) 223);
  lcd.setCursor(11, 3);
  lcd.print("C");


  if (digitalRead(ADD) == PRESSED && (temp_ref < 99))
  {
    temp_ref += 0.1;
    delay(200);
```

```
  }
  if (digitalRead(DEDUCT) == PRESSED && (temp_ref > 2))
  {
    temp_ref -= 0.1;
    delay(200);
  }
}
while (digitalRead(NEXT) == NOTPRESSED);


do
{
}
while (digitalRead(NEXT) == PRESSED); // debouncing of the menubutton pressing
lcd.setCursor(4, 1);
lcd.print("  "); // special select item character removal


do
{
  lcd.setCursor(4, 3);
  lcd.print(">>");
  lcd.setCursor(2, 2);
  lcd.print("Temp TOL Value: ");
  lcd.setCursor(6,3);
  lcd.print(temp_ref_tol);
  lcd.setCursor(10,3);
  lcd.print((char) 223);
  lcd.setCursor(11,3);
  lcd.print("C");


  if (digitalRead(ADD) == PRESSED && (temp_ref_tol < 5))
  {
    temp_ref_tol += 0.1;
    delay(200);
  }
  if (digitalRead(DEDUCT) == PRESSED && (temp_ref_tol > 0))
  {
    temp_ref_tol -= 0.1;
    delay(200);
  }
}
while (digitalRead(NEXT) == NOTPRESSED);
```

```
do
{
}
while (digitalRead(NEXT) == PRESSED);
lcd.clear();


do
{
  lcd.setCursor(4, 1);
  lcd.print(">>");
  lcd.setCursor(3, 0);
  lcd.print("Set Humidity:");
  lcd.setCursor(6, 1);
  lcd.print(humid_ref);
  lcd.setCursor(11, 1);
  lcd.print("% RH");


  lcd.setCursor(1, 2);
  lcd.print("Humidity Tolerance:");
  lcd.setCursor(6, 3);
  lcd.print(humid_ref_tol);
  lcd.setCursor(10, 3);
  lcd.print("% RH");


  if (digitalRead(ADD) == PRESSED && (humid_ref < 99))
  {
    humid_ref += 0.1;
    delay(200);
  }
  if (digitalRead(DEDUCT) == PRESSED && (humid_ref > 5))
  {
    humid_ref -= 0.1;
    delay(200);
  }
}
while (digitalRead(NEXT) == NOTPRESSED);


do
{
```

```
}
while (digitalRead(NEXT) == PRESSED); // debouncing of the menubutton pressing
lcd.setCursor(4, 1);
lcd.print("  "); // special select item character removal


do
{
  lcd.setCursor(4, 3);
  lcd.print(">>");
  lcd.setCursor(1, 2);
  lcd.print("Humidity Tolerance:");
  lcd.setCursor(6,3);
  lcd.print(humid_ref_tol);
  lcd.setCursor(10,3);
  lcd.print("% RH");


  if (digitalRead(ADD) == PRESSED && (humid_ref_tol < 99))
  {
    humid_ref_tol += 0.1;
    delay(200);
  }
  if (digitalRead(DEDUCT) == PRESSED && (humid_ref_tol > 0))
  {
    humid_ref_tol -= 0.1;
    delay(200);
  }
}
while (digitalRead(NEXT) == NOTPRESSED);


do
{
}
while (digitalRead(NEXT) == PRESSED);
lcd.clear();


do
{
  lcd.setCursor(4,1);
  lcd.print(">>");// special select item character
  lcd.setCursor(1,0);
  lcd.print("Temp_Diff Treshold:");
```

```
    lcd.setCursor(6,1);
    lcd.print(temp_ref_tres);
    lcd.setCursor(10,1);
    lcd.print((char) 223);
    lcd.setCursor(11,1);
    lcd.print("C");


    lcd.setCursor(0,2);
    lcd.print("Humid_Diff Treshold:");
    lcd.setCursor(6,3);
    lcd.print(humid_ref_tres);
    lcd.setCursor(10,3);
    lcd.print("% RH");


    if (digitalRead(ADD) == PRESSED)
    {
      temp_ref_tres += 0.1;
      delay(200);
    }
    if (digitalRead(DEDUCT) == PRESSED)
    {
      temp_ref_tres -= 0.1;
      delay(200);
    }
  }
  while(digitalRead(NEXT)== NOTPRESSED);


  do
  {
  }
  while (digitalRead(NEXT) == PRESSED); // debouncing of the menubutton pressing
  lcd.setCursor(4, 1);
  lcd.print("  "); // special select item character removal

  do
  {
    lcd.setCursor(4,3);
    lcd.print(">>");// special select item character
    lcd.setCursor(0,2);
    lcd.setCursor(0,2);
    lcd.print("Humid_Diff Treshold:");
    lcd.setCursor(6,3);
```

```
    lcd.print(humid_ref_tres);
    lcd.setCursor(10,3);
    lcd.print("% RH");

    if (digitalRead(ADD) == PRESSED)
    {
      humid_ref_tres += 0.1;
      delay(200);
    }
    if (digitalRead(DEDUCT) == PRESSED)
    {
      humid_ref_tres -= 0.1;
      delay(200);
    }
  }
  while(digitalRead(NEXT) == NOTPRESSED);


  do
  {
  }
  while (digitalRead(NEXT) == PRESSED);
  lcd.clear();


  do
  {
    if (incub_days_total > 50)
      incub_days_total = 21;
    lcd.setCursor(4,2);
    lcd.print(">>");
    lcd.setCursor(0,0);
    lcd.print("Set Incubation_Time:");
    lcd.setCursor(6,2);
    lcd.print(incub_days_total);
    lcd.setCursor(9,2);
    lcd.print("Days");

    if (digitalRead(ADD) == PRESSED && incub_days_total < 46)
    {
      incub_days_total++;
      delay(200);
    }
    if (digitalRead(DEDUCT) == PRESSED && incub_days_total > 1)
    {
```

```
      incub_days_total--;
      delay(200);
    }
    if (incub_days_total < 10)
    {
      lcd.setCursor(7, 2);
      lcd.print(" ");
    }
  }
  while (digitalRead(NEXT) == NOTPRESSED);


  do
  {
  }
  while (digitalRead(NEXT) == PRESSED);
  lcd.clear();


  do
  {
    lcd.setCursor(4, 2);
    lcd.print(">>");
    lcd.setCursor(0, 0);
    lcd.print("Time Between Spins:");
    lcd.setCursor(6,2);
    lcd.print(SPIN_time);
    lcd.setCursor(9,2);
    lcd.print("Hour/s");


    if (digitalRead(ADD) == PRESSED && SPIN_time < 13)
    {
      SPIN_time++;
      delay(100);
    }
    if (digitalRead(DEDUCT) == PRESSED && SPIN_time > 0)
    {
      SPIN_time--;
      delay(100);
    }
    if (SPIN_time < 10)
    {
      lcd.setCursor(7, 2);
      lcd.print(" ");
```

```
    }
  }
  while (digitalRead(NEXT) == NOTPRESSED);


  do
  {
  }
  while (digitalRead(NEXT) == PRESSED);
  lcd.clear();


  do
  {
    lcd.setCursor(4, 2);
    lcd.print(">>");
    lcd.setCursor(0, 0);
    lcd.print("Time During Rot:");
    lcd.setCursor(6, 2);
    lcd.print(rotation_duration);
    lcd.setCursor(8, 2);
    lcd.print("s");


    if (rotation_duration < 10)
    {
      lcd.setCursor(7, 2);
      lcd.print(" ");
    }
    if (digitalRead(ADD) == PRESSED && rotation_duration < 100)
    {
      rotation_duration++;
      if (rotation_duration < 10)
      {
        lcd.setCursor(7, 2);
        lcd.print(" ");
      }
      delay(200);
    }
    if (digitalRead(DEDUCT) == PRESSED && rotation_duration > 1)
    {
      rotation_duration--;
      // if (rotation_duration < 10)
      // {
        lcd.setCursor(7, 2);
```

```
        lcd.print(" ");
      // }
      delay(200);
    }
  }
  while (digitalRead(NEXT) == NOTPRESSED);


  do
  {
  }
  while (digitalRead(NEXT) == PRESSED);
  lcd.clear();


  do
  {
    lcd.setCursor(0, 0);
    lcd.print("Press 'NEXT' to save");
    lcd.setCursor(2,1);
    lcd.print(" any changes or");
    lcd.setCursor(0,2);
    lcd.print("any key to continue");
  }
  while (digitalRead(NEXT) == NOTPRESSED && digitalRead(ADD) == NOTPRESSED &&
digitalRead(DEDUCT) == NOTPRESSED);


  if (!digitalRead(NEXT))
  {
    lcd.clear();
    ft_save_param();


    lcd.clear();
    lcd.print("SAVING...");
    lcd.setCursor(3, 3);
    for (int i = 0; i < 17; i++)
    {
      lcd.setCursor(i + 1, 1);
      lcd.write(byte(10));
      lcd.setCursor(i + 1, 2);
      lcd.write(byte(10));
      delay(60);
    }
```

```
    lcd.clear();
    lcd.setCursor(0, 1);
    lcd.print("SAVED SUCCESSFULLY!!");
    delay(500);
  }
  else
  {
    lcd.clear();
    ft_discard_param();
    lcd.setCursor(0,0);
    lcd.print("changes discarded");
    delay(300);
  }
  lcd.clear();
}


void  loop()
{
  now = rtc.now();

  lcd.setCursor(15, 2);
  lcd.print(SPIN_second);
  lcd.setCursor(0, 0);
  lcd.print("Temp= ");
  lcd.setCursor(6, 0);
  lcd.print(temperature);
  espSerial.print(" "); // Serial Communication //
  espSerial.print(temperature); // Serial Communication //
  delay(200);
  espSerial.print(","); // Serial Communication //
  espSerial.print(humidity); // Serial Communication //
  delay(200);
  lcd.setCursor(12, 0);
  lcd.print((char) 223);
  lcd.setCursor(13, 0);
  lcd.print("C");
  lcd.setCursor(0, 1);
  lcd.print("Humid= ");
  lcd.setCursor(7, 1);
  lcd.print(humidity);
  lcd.setCursor(13, 1);
  lcd.print("%");
```

```
if (SPIN_time == 0)
{
  lcd.setCursor(0, 2);
  lcd.print("RotTime= ");
  lcd.setCursor(9, 2);
  lcd.print("00");
  lcd.setCursor(11, 2);
  lcd.print(":");
  lcd.setCursor(12, 2);
  lcd.print("00");
  lcd.setCursor(14, 2);
  lcd.print(":");
  lcd.setCursor(15, 2);
  lcd.print("00");
}
else
{
  lcd.setCursor(0, 2);
  lcd.print("RotTime= ");
  lcd.setCursor(9, 2);
  lcd.print(SPIN_hour);
  espSerial.print("!"); // Serial Communication //
  espSerial.print(SPIN_hour); // Serial Communication //
  delay(200);
  lcd.setCursor(11, 2);
  lcd.print(":");
  lcd.setCursor(12, 2);
  lcd.print(SPIN_minute);
  espSerial.print("@"); // Serial Communication //
  espSerial.print(SPIN_minute); // Serial Communication //
  delay(200);
  lcd.setCursor(14, 2);
  lcd.print(":");
  lcd.setCursor(15, 2);
  lcd.print(SPIN_second);
  espSerial.print("#"); // Serial Communication //
  espSerial.print(SPIN_second); // Serial Communication //
  delay(200);
}


lcd.setCursor(0, 3);
lcd.print("CountDown= ");
lcd.setCursor(11, 3);
lcd.print(incub_days_left);
```

```
espSerial.print("$"); // Serial Communication //
espSerial.print(incub_days_left); // Serial Communication //
delay(200);


if (incub_days_left < 10)
{
  lcd.setCursor(12, 3);
  lcd.print(" ");
}
lcd.setCursor(14, 3);
lcd.print("Days");


lcd.setCursor(15, 2);
lcd.print(SPIN_second);
if (SPIN_second < 10)
{
  lcd.setCursor(16, 2);
  lcd.print(" ");
}
if (SPIN_minute < 10)
{
  lcd.setCursor(13, 2);
  lcd.print(" ");
}
if (SPIN_hour < 10)
{
  lcd.setCursor(10, 2);
  lcd.print(" ");
}


if (SPIN_second == 0 && SPIN_minute)
{
  SPIN_minute -= 1;
  SPIN_second = 59;
}
if (SPIN_minute == 0 && SPIN_hour)
{
  SPIN_hour -= 1;
  SPIN_minute = 59;
}
if (SPIN_second == 0 && SPIN_minute == 0 && SPIN_hour == 0)
{
```

```
    delay(20);
    motor_direction = EEPROM.read(motor_dir_address);
    delay(20);
    espSerial.print("*"); // Serial Communication //
    espSerial.print(1); // Serial Communication //
    delay(200);
    if (motor_direction)
    {
      motor_direction = 0;
      delay(20);
      EEPROM.write(motor_dir_address, motor_direction);
      delay(20);
      lcd.clear();
      lcd.setCursor(0, 1);
      lcd.print("TRAY ROTATE RIGHT");
      digitalWrite(TRAY_LEFT, LOW);
      digitalWrite(TRAY_RIGHT, HIGH);
      delay(rotation_duration * 1000);
      digitalWrite(TRAY_RIGHT, LOW);
      digitalWrite(TRAY_LEFT, LOW);
      lcd.clear();
    }
    else
    {
      motor_direction = 1;
      delay(20);
      EEPROM.write(motor_dir_address, motor_direction);
      delay(20);
      lcd.clear();
      lcd.setCursor(0, 1);
      lcd.print("TRAY ROTATE LEFT");
      digitalWrite(TRAY_RIGHT, LOW);
      digitalWrite(TRAY_LEFT, HIGH);
      delay(rotation_duration * 1000);
      digitalWrite(TRAY_RIGHT, LOW);
      digitalWrite(TRAY_LEFT, LOW);
      lcd.clear();
    }
    SPIN_time = EEPROM.read(SPIN_time_address);
    SPIN_hour = SPIN_time - 1;
    SPIN_minute = 59;
    SPIN_second = 59;
    ft_reset();
  }
  else
```

```
  {
    espSerial.print("*"); // Serial Communication //
    espSerial.print(0); // Serial Communication //
    delay(200);
  }
  lcd.setCursor(15, 2);
  lcd.print(SPIN_second);


  ft_temp_regulation();
  ft_humid_regulation();


  if ((temperature >= (temp_ref + temp_ref_tol + temp_ref_tres)) || (temperature <= (temp_ref
- temp_ref_tol - temp_ref_tres - 0.1))
    || (humidity >= (humid_ref + humid_ref_tol + humid_ref_tres + 0.1)) || (humidity <=
(humid_ref - humid_ref_tol - humid_ref_tres - 0.1)))
  {
    espSerial.print("`"); // Serial Communication //
    espSerial.print(1); // Serial Communication //
    delay(200);
  }
  else
  {
    espSerial.print("`"); // Serial Communication //
    espSerial.print(0); // Serial Communication //
    delay(200);
  }


  lcd.setCursor(15, 2);
  lcd.print(SPIN_second);


  if (digitalRead(NEXT) == PRESSED)
    ft_menu();
  lcd.setCursor(15, 2);
  lcd.print(SPIN_second);


  if (!incub_days_left)
  {
    ft_buzzer();
    lcd.setCursor(0, 3);
    lcd.print("                    ");
```

```
    delay(10);
    lcd.setCursor(0, 3);
    lcd.print("!!! EGG HATECHED !!!");
    delay(500);
    lcd.setCursor(0, 3);
    lcd.print("                    ");
    delay(10);
    espSerial.print("?"); // Serial Communication //
    espSerial.println(1); // Serial Communication //
    delay(200);
  }
  else
  {
    espSerial.print("?"); // Serial Communication //
    espSerial.println(0); // Serial Communication //
    delay(200);
  }
  lcd.setCursor(15, 2);
  lcd.print(SPIN_second);


  DateTime startday(a, b, c, d, e, f);
  delay(20);


/* This code from function below is for demonstration which allows the day value to go in
seconds instead of seconds */
  // while (n < 50)
  // {
  //   if (incub_days_total >= n)
  //   {
  //     if (rtc.now() == (startday + TimeSpan(0, 0, 0, n)))
  //     {
  //       incub_days_left = incub_days_total - n;
  //       EEPROM.write(incub_days_left_address, incub_days_left);
  //       delay(10);
  //     }
  //   }
  //   n++;
  // }


/*  This code from function below is the proper day duration instead of seconds */
```

```
  for (unsigned short n=1; n<50; n++)
  {
    if (incub_days_total >= n)
    {
      if  ((rtc.now() == ( startday + TimeSpan(n,0,0,0)))||(rtc.now() == ( startday +
TimeSpan(n,0,30,0)))||(rtc.now() == ( startday + TimeSpan(n,1,0,0)))||(rtc.now() == ( startday
+ TimeSpan(n,2,0,0)))||(rtc.now() == ( startday + TimeSpan(n,5,0,0)))||(rtc.now() == (
startday + TimeSpan(n,10,0,0))))
      {
        incub_days_left = incub_days_total - n;
        EEPROM.write(incub_days_left_address, incub_days_left);
        delay(10);
        }
    }
}


  // while (n < 50)
  // {
  //   if (incub_days_total >= n)
  //   {
  //     if (rtc.now() == (startday + TimeSpan(n, 0, 0, 0)) || rtc.now() == (startday +
TimeSpan(n, 0, 30, 0)) || rtc.now() == (startday + TimeSpan(n, 1, 0, 0))
  //       || rtc.now() == (startday + TimeSpan(n, 2, 0, 0)) || rtc.now() == (startday +
TimeSpan(n, 5, 0, 0)) || rtc.now() == (startday + TimeSpan(n, 10, 0, 0)))
  //     {
  //       incub_days_left = incub_days_total - n;
  //       EEPROM.write(incub_days_left_address, incub_days_left);
  //       delay(10);
  //     }
  //   }
  //   n++;
  // }
}
```

ESP8266 Software Codes

```
#include <SPI.h>
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>


#define BLYNK_PRINT Serial
#define BLYNK_TEMPLATE_ID "TMPL6i5aLKSsY"
```

```
#define BLYNK_TEMPLATE_NAME "Egg Incubator"
#define BLYNK_AUTH_TOKEN "csTti0NEHCdKXKjlKyFVMz1JCLQwcA1l"
#define RESET D1


WidgetLED Bulb(V7);
WidgetLED Humidifier(V8);
WidgetLED Fan(V9);
WidgetLED Motor(V10);
WidgetLED Buzzer(V11);
WidgetLCD lcd(V12);


char auth[] = "csTti0NEHCdKXKjlKyFVMz1JCLQwcA1l";
char ssid[] = "Ben";
char pass[] = "11111111";


BLYNK_WRITE(V14)
{
  int pinValue = param.asInt();
  digitalWrite(RESET, pinValue);
}


void setup()
{
  digitalWrite(RESET, HIGH);
  Serial.begin(9600);
  Serial1.begin(115200);
  pinMode(RESET, OUTPUT);
  Blynk.begin(auth, ssid, pass);
  while (!Serial)
  {
    ; // wait for serial port to connect. Needed for native USB port only
  }
}


void loop()
{
  char message_1[14]; // Array to store the formatted message
  char message_2[8];
  sprintf(message_1, "Not Yet Hatch");
  sprintf(message_2, "Hatched");
```

```
  // run over and over
  if (Serial.available())
  {
    Serial.write(Serial.read());
    String receivedData = Serial.readStringUntil('\n');
    delay(100);
    float Index_1 = receivedData.indexOf(',');
    float Index_2 = receivedData.indexOf('!');
    float Index_3 = receivedData.indexOf('@');
    float Index_4 = receivedData.indexOf('#');
    float Index_5 = receivedData.indexOf('$');
    float Index_6 = receivedData.indexOf('*');
    float Index_7 = receivedData.indexOf('%');
    float Index_8 = receivedData.indexOf('^');
    float Index_9 = receivedData.indexOf('&');
    float Index_10 = receivedData.indexOf('`');
    float Index_11 = receivedData.indexOf('?');


      if (Index_1 != -1 || Index_2 != -1 || Index_3 != -1 || Index_4 != -1 || Index_5 != -1
        || Index_6 != -1 || Index_7 != -1 || Index_8 != -1 || Index_9 != -1 || Index_10 != -1
|| Index_11 != -1)
      {
        float receivedTemperature = receivedData.substring(0, Index_1).toFloat();
        float receivedHumidity = receivedData.substring(Index_1 + 1, Index_2).toFloat();
        unsigned short receivedHour = receivedData.substring(Index_2 + 1, Index_3).toInt();
        unsigned short receivedMinute = receivedData.substring(Index_3 + 1, Index_4).toInt();
        unsigned short receivedSecond = receivedData.substring(Index_4 + 1, Index_5).toInt();
        unsigned short receivedDay = receivedData.substring(Index_5 + 1, Index_6).toInt();
        unsigned short receivedMotor = receivedData.substring(Index_6 + 1, Index_7).toInt();
        unsigned short receivedBulb = receivedData.substring(Index_7 + 1, Index_8).toInt();
        unsigned short receivedFan = receivedData.substring(Index_8 + 1, Index_9).toInt();
        unsigned short receivedHumidifier = receivedData.substring(Index_9 + 1,
Index_10).toInt();
        unsigned short receivedBuzzer = receivedData.substring(Index_10 + 1,
Index_11).toInt();
        unsigned short receivedSentence = receivedData.substring(Index_11 + 1).toInt();
        // Do something with the received values
        Blynk.run(); // Initiates Blynk
          delay(1000);
          Blynk.virtualWrite(V6, receivedTemperature);
          Blynk.virtualWrite(V5, receivedHumidity);
          Blynk.virtualWrite(V0, receivedHour);
          Blynk.virtualWrite(V1, receivedMinute);
          Blynk.virtualWrite(V2, receivedSecond);
```

```
        Blynk.virtualWrite(V3, receivedDay);
      if (receivedMotor == 1)
      Motor.on();
      else
        Motor.off();
      if (receivedBulb == 1)
        Bulb.on();
      else
        Bulb.off();
      if (receivedFan == 1)
        Fan.on();
      else
        Fan.off();
      if (receivedHumidifier == 1)
        Humidifier.on();
      else
        Humidifier.off();
      if (receivedBuzzer == 1)
        Buzzer.on();
      else
        Buzzer.off();
      if (receivedSentence == 1)
      {
        lcd.clear();
        lcd.print(0,0,message_2);
      }
      else
      {
        lcd.clear();
        lcd.print(0,0,message_1);
      }
      delay(1000);
      lcd.clear();
      }
      Serial.println(receivedData);
    }
}
```