

**MOTORIZED WHEELBARROW**

**AMIRUL FIRDAUS BIN ABU SAMAH**

**BACHELOR OF ENGINEERING TECHNOLOGY  
(ELECTRICAL) WITH HONOURS.**

**UNIVERSITI MALAYSIA PAHANG**

## UNIVERSITI MALAYSIA PAHANG

### DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : AMIRUL FIRDAUS BIN ABU SAMAH  
Date of Birth :  
Title : MOTORIZED WHEELBARROW  
Academic Session : 2020/2021

I declare that this thesis is classified as:

- CONFIDENTIAL (Contains confidential information under the Official Secret Act 1997)\*
- RESTRICTED (Contains restricted information as specified by the organization where research was done)\*
- OPEN ACCESS I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

\_\_\_\_\_  
(Student's Signature)

AMIRUL FIRDAUS

Date:

\_\_\_\_\_  
(Supervisor's Signature)

PM Ts. DR HADI MANAP

Date:

NOTE : \* If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.



## **SUPERVISOR'S DECLARATION**

I hereby declare that I have checked this thesis and, in my opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Bachelor of Engineering Technology in Electrical

Full Name : PM DR HADI BIN MANAP

Position : LECTURER

Date :

---

(Co-supervisor's Signature)

Full Name :

Position :

Date :



## **STUDENT'S DECLARATION**

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

---

(Student's Signature)

Full Name : AMIRUL FIRDAUS BIN ABU SAMAH

ID Number : TB17045

Date :

MOTORIZED WHEELBARROW

AMIRUL FIRDAUS BIN ABU SAMAH

Thesis submitted in fulfillment of the requirements  
for the award of the degree of  
Bachelor of Engineering Technology in Electrical

Faculty of Engineering Technology Electrical & Electronics  
UNIVERSITI MALAYSIA PAHANG

JANUARY 2021

## **ACKNOWLEDGEMENTS**

I am very grateful to ALLAH "S.W. T" for giving me knowledge, endurance, patience and assistance in completing my entire project. This study might not have been feasible if it had not been in His will and favor.

I would like to thank my esteemed supervisor Dr Hadi Manap for his invaluable supervision, support and instruction during the course of my senior year project. My gratitude extends to the Faculty of Engineering Technology Electrical and Electronic for prepared and supplying our project with tools and facilities to work. Also thank to En. Wan Hassan and other JP that involve in helping directly or indirectly throughout this project. Besides, also thank to my project partner, Yap Soon Guan for support each other when doing this project and finish it together, without him this project will take longer to finish.

Lastly, I appreciate all the support I received from the rest of my family, especially my mother. Without her tremendous encouragement, it would be impossible for me to complete this project.

## **ABSTRAK**

Tesis ini memperkenalkan kereta sorong bermotor yang bertujuan untuk meningkatkan reka bentuk kereta sorong konvensional. Ia terdiri daripada kereta sorong, motor hub, penggerak linear dan sensor berat. Motor hub digunakan untuk memudahkan pergerakan naik dan turun cerun. Ia menggunakan aktuator linear untuk memudahkan proses pemungahan. Bacaan berat dilakukan oleh sensor berat dengan menggunakan Arduino UNO sebagai mikrokontroler dalam system ini. Ia juga menggunakan tiga roda untuk menstabilkan kereta sorong ketika bergerak ke depan, ke belakang atau berputar kea rah berlainan. Pemacu motor PWM digunakan sebagai pengawal yang mengawal kelajuan dan arah putaran motor.

## **ABSTRACT**

This thesis introduces a motorized wheelbarrow aimed at improving the design of the conventional wheelbarrow. It consists of wheelbarrow, hub motor, linear actuator and weight sensor. The hub motor is used to improve the wheelbarrow handling that can move it easily when going up or down the slope. It uses a linear actuator to ease the unloading process. The reading of the weight is measured and detected by weight sensor and Arduino UNO as a microcontroller in this system. It using 3 tires to stable the wheelbarrow when moving forward, backward or rotating the other direction. The PWM motor driver as a controller that control the speed and direction of rotation of motor.



## TABLE OF CONTENT

<b>DECLARATION</b>	
<b>TITLE PAGE</b>	
<b>ACKNOWLEDGEMENTS</b>	<b>ii</b>
<b>ABSTRAK</b>	<b>iii</b>
<b>ABSTRACT</b>	<b>iv</b>
<b>TABLE OF CONTENT</b>	<b>v</b>
<b>LIST OF TABLES</b>	<b>viii</b>
<b>LIST OF FIGURES</b>	<b>ix</b>
<b>LIST OF SYMBOLS</b>	<b>x</b>
<b>LIST OF ABBREVIATIONS</b>	<b>xi</b>
<b>LIST OF APPENDICES</b>	<b>xii</b>
<b>CHAPTER 1 INTRODUCTION</b>	<b>1</b>
1.1 Project background	1
1.2 Problem Statement	2
1.3 Objectives	3
1.4 Project Scope	3
<b>CHAPTER 2 LITERATURE REVIEW</b>	<b>4</b>
2.1 Introduction	4
2.2 Background	4
2.3 Conventional Wheelbarrow	5
2.4 Powered Wheelbarrow	6
2.5 Motorized Wheelbarrow	7
2.6 Linear Actuator	8

2.6.1	Linear Actuator Application	8
2.7	Weight Sensor	10
2.7.1	Type of Weight Sensor	10
<b>CHAPTER 3 METHODOLOGY</b>		<b>11</b>
3.1	Introduction of Methodology	11
3.2	System Overview	13
3.3	Weight Sensor Operation	15
3.4	Project Flow	16
3.5	Wiring and Soldering	17
3.6	Material	18
3.6.1	Battery	18
3.6.2	Linear Actuator	19
3.6.3	MD10C R3 Motor Driver	20
3.6.4	Arduino UNO	22
3.6.5	Buzzer	23
3.6.6	Weight Sensor Strain Gauge	24
3.6.7	HX711	26
<b>CHAPTER 4 RESULTS AND DISCUSSION</b>		<b>27</b>
4.1	Introduction	27
4.2	Weight Sensor Strain Gauge Connection	27
4.3	Calibration	29
4.4	Linear Actuator	31
4.5	Wiring Connection	33

<b>CHAPTER 5 CONCLUSION</b>	<b>34</b>
5.1 Conclusion	34
5.2 Recommendation	34
<b>REFERENCES</b>	<b>35</b>
<b>APPENDICES</b>	<b>36</b>

## LIST OF TABLES

Table 1: Battery state of charge

18

## LIST OF FIGURES

Figure 1.1: Example of Existing Wheelbarrow	2
Figure 2.1: Conventional Wheelbarrow	5
Figure 2.2: Powered Wheelbarrow	6
Figure 2.3: Motorized Wheelbarrow	7
Figure 2.4: Example of weight sensor	10
Figure 3.1: Methodology Process Flowchart	12
Figure 3.2: Flowchart of The System Run	14
Figure 3.3: Weight Sensor Operation	15
Figure 3.4: Motorized Wheelbarrow System Block Diagram	16
Figure 3.5: Soldering	17
Figure 3.6: Heavy Duty Linear Actuator	19
Figure 3.7: MD10C R3 PWM Motor Driver	20
Figure 3.8: Arduino UNO R3	22
Figure 3.9: Buzzer	23
Figure 3.10: Weight Sensor Strain Gauge	25
Figure 3.11: HX711	26
Figure 4.1: Weight sensor	27
Figure 4.2: Setup of weight sensor	28
Figure 4.3: Calibrating weight	30
Figure 4.4: Speed of linear actuator	31
Figure 4.5: Speed against weight of load	32

## LIST OF SYMBOLS

V	Voltage
W	Watt
N	Newton
%	Percentage
A	Ampere
g	Gram
+	Plus
-	Minus
<	Less than

## LIST OF ABBREVIATIONS

PWM	Pulse Width Modulation
kg	Kilogram
DC	Direct current
O/P	Output
LCD	Liquid Crystal Display
IDE	Integrated Development Environment
mm/s	Millimetre per second
kHz	Kilo Hertz
LED	Light Emitting Diode
dB	Decibel
GND	Ground
I/O	Input Output
ADC	Analog-to-digital converter
mA	Milliampere
PVC	Polyvinyl chloride

**LIST OF APPENDICES**

APPENDIX A 37  
APPENDIX B 43  
APPENDIX C 47  
APPENDIX D 51  
APPENDIX E 52  
APPENDIX F 53  
APPENDIX G 54  
APPENDIX H 55



# CHAPTER 1

## INTRODUCTION

### 1.1 Project background

Wheelbarrow has become an important part of the daily routine of human beings in carrying heavy loads, especially often used in agriculture. This allows us to move items from one position to another without using as much energy as we bring by hand. Nevertheless, the restriction is always to transport it by climbing up the hill with heavy loads. The motorized wheelbarrow is a safe and suitable to be used to climb up and down slopes.

This motorized wheelbarrow feature of linear actuator and electric motor. The electric hub motor works to produce force in the form of shaft rotation by communicating with the magnetic field of the motor and electric current in a wire winding. The electric hub motor can be driven via a throttle grip at the handle and generate power to the wheelbarrow. The linear actuator is an actuator that, in a straight line, generates motion that could be used to land heavy loads. In addition, this motorized wheelbarrow added two additional tyres to the back of the wheelbarrow. While climbing up and down the hill, the three-wheel wheelbarrow gives the heavy loads more stability, to change directions, it is also able to rotate 360 degree easily.

This project focuses on the product's problem solving and offers specific mechanisms and technologies used to invent a motorized wheelbarrow that carry heavy objects and unloading the object.

## 1.2 Problem Statement

The wheelbarrow (Figure 1.1), design can be physically challenging to the user when performing tasks where they have to bend forward or pull a load back up a slope, this can cause the user to strain their back. From a design aspect, the wheelbarrow is a very simple concept that facilitates the transportation of large amounts of difficult to move materials. In the construction and gardening industries, it is most widely used to transfer such materials as rocks, gravel, stones, plants, concrete and many more.

One concern that occurs when using this piece of equipment is that there only one way to unload the item, by a load-removing bar placed in front of the wheel, which serves as a lever when unloading bucket contents. The wheelbarrow will continue to move forward without the pivot bar while trying to empty the bucket, tipping the bucket in turn becomes more strenuous. There is no mechanism system to lift the bucket to unload the weight.

Another problem is that the traditional wheelbarrow consists of one wheel, because it is incapable of holding a large amount of weight and is difficult to stabilize the weight. As the weight increase, it will become harder to control the direction of the wheelbarrow because of the design that uses one wheel in front of the wheelbarrow.

The wheelbarrow has to be managed by using huge quantities of energy to climb up and down the slope. Besides, the wheelbarrow does not provide a brake so that the person using it will have a hard time controlling the wheelbarrow on the decline slope.



**Figure 1.1:** Example of Existing Wheelbarrow

### **1.3 Objectives**

The study is focused on how to make traditional wheelbarrow to become motorized wheelbarrow. The scopes of our study were as follows:

1. To develop a wheelbarrow that utilize linear actuator to unload the wheelbarrow.
2. To make the wheelbarrow more stable when handling using three tires.
3. To make weight sensor as safety measured when unload the wheelbarrow

### **1.4 Project Scope**

The main study or analysis of this project is:

1. Implementation of linear actuator that could sustain up to 100kg.
2. Implementation of three-wheel wheelbarrow that make wheelbarrow easier to rotate and changing direction.
3. Implementation of weight sensor to stop actuator go beyond 100kg for safety measured.

## **CHAPTER 2**

### **LITERATURE REVIEW**

#### **2.1 Introduction**

This chapter provides a literature review that is connected to this project. From the history of this project and previous system, it explains more information about subtopics. Based on previous studies, it is possible to compare and develop this project.

#### **2.2 Background**

The project is intended to upgrade the traditional wheelbarrows that are commonly used in our everyday lives. Wheelbarrows were usually used for the garden in the past. Gardeners use wheelbarrows to fill and move heavy loads from one position to another, such as soils. However, people use them these days for a number of purposes such as at home, in the garden, in building, stonework, roadwork and more.

As the use of wheelbarrows continues to increase today, by incorporating electrical system and motorized system, the regular wheelbarrow is undergoing several innovations. The benefit of using the motorized wheelbarrow is that the pressure on the back can be minimized as it does not take a lot of power to lift the tray as a linear actuator has been used. Using a motorized wheelbarrow will save time and faster job done than traditional wheelbarrows with less back and forth trips needed. The motorized wheelbarrow was developed because it can support human by reducing human energy consumption.

### **2.3 Conventional Wheelbarrow**

There are several similar products related to this current project based on the research that has done. A conventional wheelbarrow is a small hand-powered vehicle with usually only one wheel and two legs built to be pushed and driven by a single person using two handles at the back. The wheelbarrow is designed to divide the weight of its load between the wheel and the operator, allowing for the comfortable carrying of heavier and bulkier loads which would have been necessary if the operator were to bear the weight entirely.

As state in US patent 5,924,708 which issued to Bisailon et al on Jul. 20, 1999, there is difficulty in steering the conventional wheelbarrow when filling with a heavy load. This becomes apparent when the load increase. This can lead the wheelbarrow to tip when turning the wheelbarrow. Figure 2.1 show conventional wheelbarrow.



**Figure 2.1:** Conventional Wheelbarrow

## 2.4 Powered Wheelbarrow

Another one of the projects known as “Powered Wheelbarrow” by Kevin Wagner. There are some enhancements and shortcomings we need to improvise on his project after studying, evaluating, and modifying his project. In this project, scissor jack. Black Bull Electric Car Jack was used as a lifting system. This jack runs off a 12V battery supply and is capable of lifting 1-tonne weight. The wheelbarrow drive system uses a scooter motor that been aligned with the chain and sprocket to move. The design of the wheelbarrow is almost similar to a normal wheelbarrow that uses one wheel at the front. Figure 2.2 show powered wheelbarrow.

However, the weakness of this project shown when the electric car jack can only lift the load for the first 9 inches, and the remainder of the tilting needs to be done by human intervention by the handle at the back of the tray. Although the wheelbarrow uses motor to move, the operator still needs to lift the back legs off of the ground to allow the motor to drive the load.

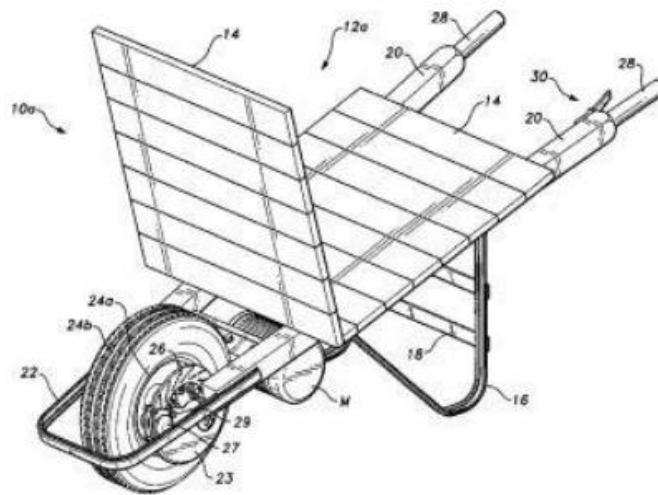


**Figure 2.2:** Powered Wheelbarrow

## 2.5 Motorized Wheelbarrow

Besides that, in “Motorized Wheelbarrow” US patent 8,235,153 which issued to Robinson et al Aug. 7, 2012, The wheelbarrow has longitudinally extended frame components which extend into portions of the handle. The wheelbarrow rest height change is accomplished with a pair of tubular legs jointly attached to a wheelbarrow support strut and a support strut brace. It also uses a hydraulic disk braking system for braking. Under the support frame, a battery is mounted next to the motor. This design has a good braking system for stopping the wheelbarrow with a big load because the disk brake has a disk rotor and calliper to control for safe wheelbarrow stopping.

However, it has a flaw where the motor itself is attached almost near to the ground. This can damage the motor when the wheelbarrow moves on uneven ground. The motor has the potential to hit a rock or a wood log during use. The chain also can come off as a result. The tray or bucket of the wheelbarrow also has to be tilt with the whole wheelbarrow. This can make the user feel more burden because additional weight of the motor. Figure 2.3 shows the design of the motorized wheelbarrow.



**Figure 2.3:** Motorized Wheelbarrow

## 2.6 Linear Actuator

An electric linear actuator is a device that transforms a low voltage DC motor's rotational motion into linear movement that pushes and pulls movements. This way, you can simply press a button to lift, adjust, rotate, push or pull heavy or difficult-to-reach objects.

Electrical linear actuators provide precise motion control with safe, quiet and clean motion. It is energy efficient with little to no maintenance and has a long lifespan. Compared to hydraulic systems, building an actuator is really simple because it takes up far less room since it has no pumps or tubing.

### 2.6.1 Linear Actuator Application

1. **Material handling.** This is a universal need for every type of manufacturing operation.
2. **Robotics.** The automotive industry and any number of others are now using robotics to improve production quality and accuracy and control production costs. Electric linear actuators meet the sophisticated needs of robotics. They can control and repeat extremely precise movements, control rate of acceleration and deceleration, and control the amount of force applied. And they can combine all these movements on multiple axes simultaneously.
3. **Food and beverage manufacturing.** Cleanliness is critical in these industries, and electric linear actuators are both clean and quiet. In addition, food and beverage, medical device, semiconductor, and some other applications also require stringent washdown protocols. Electric actuators are corrosion-resistant and have a smooth design that offers few crevices where bacteria or dirt might accumulate.



- 4. Window automation.** Manufacturing facilities and other large-scale indoor operations are constructed with heavy-duty ventilation systems, but in some cases, natural ventilation is also desirable, especially to help control indoor temperature. Electric linear actuators make it easy to remotely open and close heavy and/or high windows.
- 5. Agricultural machinery.** Although heavy equipment and attachments are often powered with hydraulics, machines that directly contact food or which require finessed movements can be fitted with electrical actuators instead. Examples include combines that thresh and convey grains, spreaders with adjustable nozzles, and even tractors.
- 6. Solar panel operation.** For optimal operation, solar panels must tilt to directly face the sun as it moves across the sky. Electric actuators enable commercial installations and utilities to efficiently and consistently control large solar farms.

## 2.7 Weight Sensor

The function of this sensor is converting a load into an electrical signal. The signal can be a change in voltage; current otherwise frequency based on the load as well as used circuit. Technically, this sensor senses changes like force, vibration or weight inside a physical stimulus and generates an output that is comparable to the physical stimulus. So, this sensor provides an output value for a particular stable load otherwise weight size, and that is comparable to the magnitude of the weight.



**Figure 2.4:** Example of weight sensor

### 2.7.1 Type of Weight Sensor

With the help of a single load cell, there are several options available for measuring a stable weight. Load cells or weight sensors are divided into various groups that are used in many applications for weight measurement. The strain gauge, capacitance, hydraulic and pneumatic are the most commonly used weight sensors. The weight sensor we are using in this project is a weight sensor of the strain gauge kind.

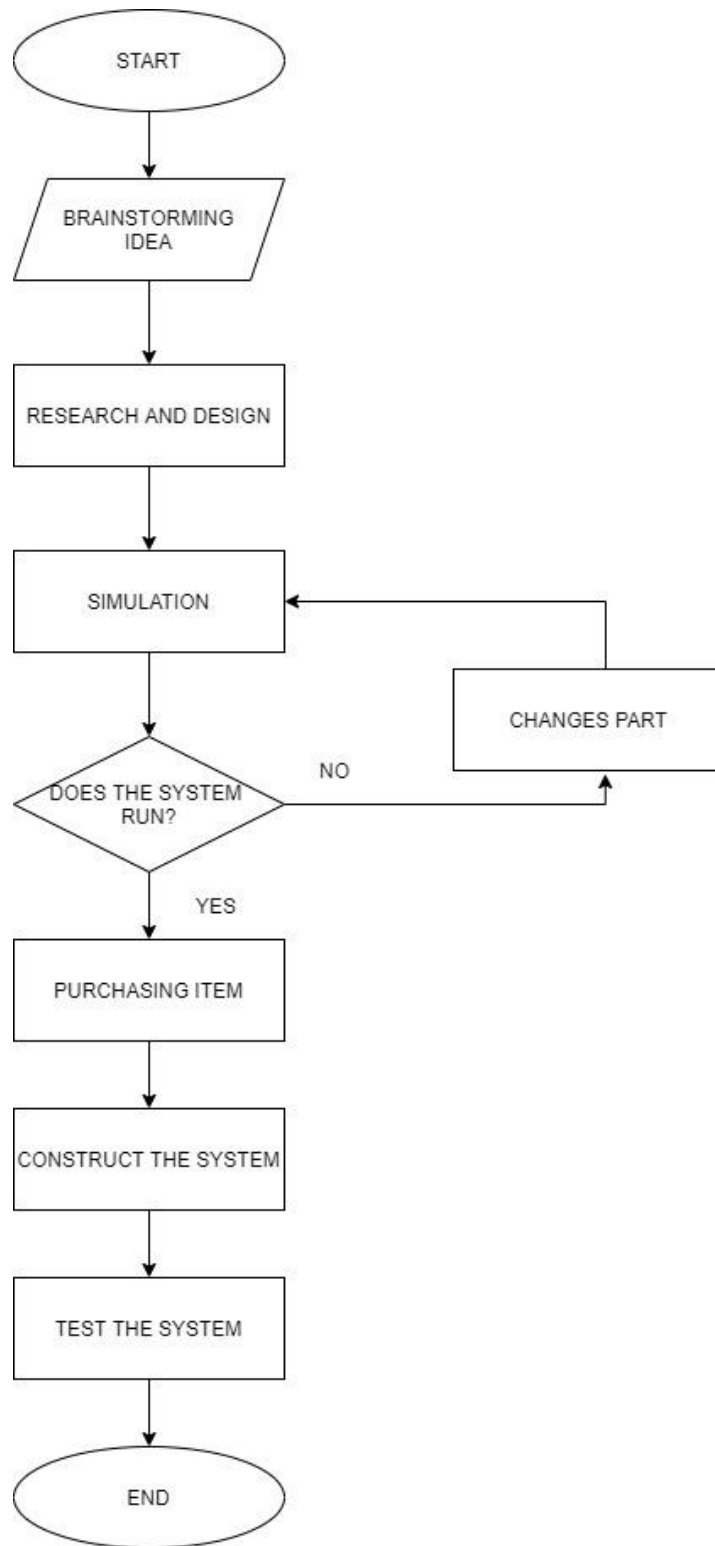
The first two are electrical transducer devices for the above-mentioned sensor types. It is a sensor used as an output to sense a physical stimulus and to produce voltage. The remaining two sensors do not produce output openly as electrical signals, but depending on the application situation, they may have their O/P. In various industries, such as domestic, robotics, medicine, and automotive, the strain gauge is the most widely used sensor.

## **CHAPTER 3**

### **METHODOLOGY**

#### **3.1 Introduction of Methodology**

This chapter discusses the development of hardware and software, such as Motorized Wheelbarrow equipment, coding, processes and method design. The applicable data is collected from the previous chapter via a literature review. This chapter will also cover part-by-part circuits and full circuits. It is best to begin with a quick overview of the architecture of the system before looking at the specifics of all the methods below. On Figure 3.1, the technique presents the methods in the flow chart.

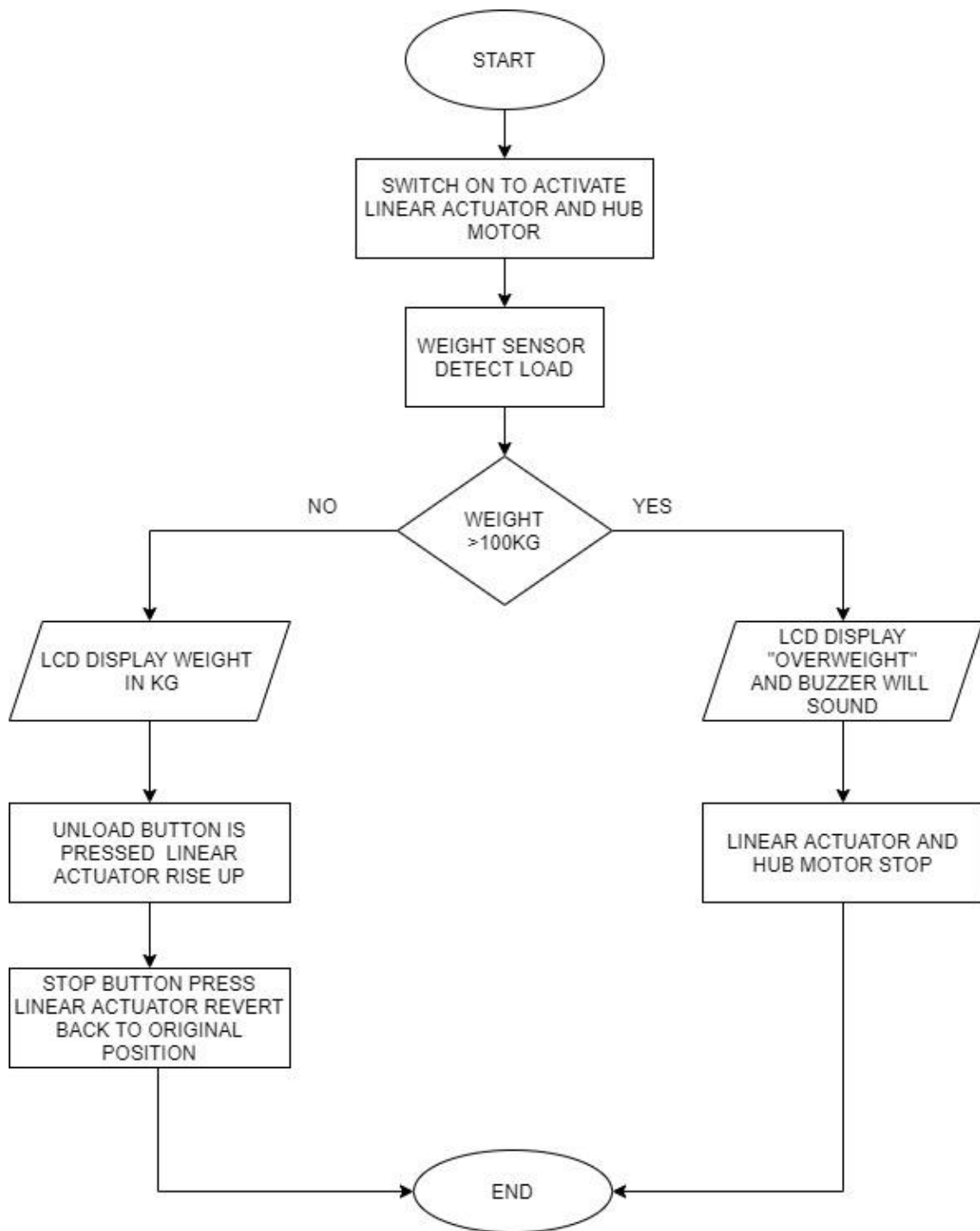


**Figure 3.1:** Methodology Process Flowchart

### 3.2 System Overview

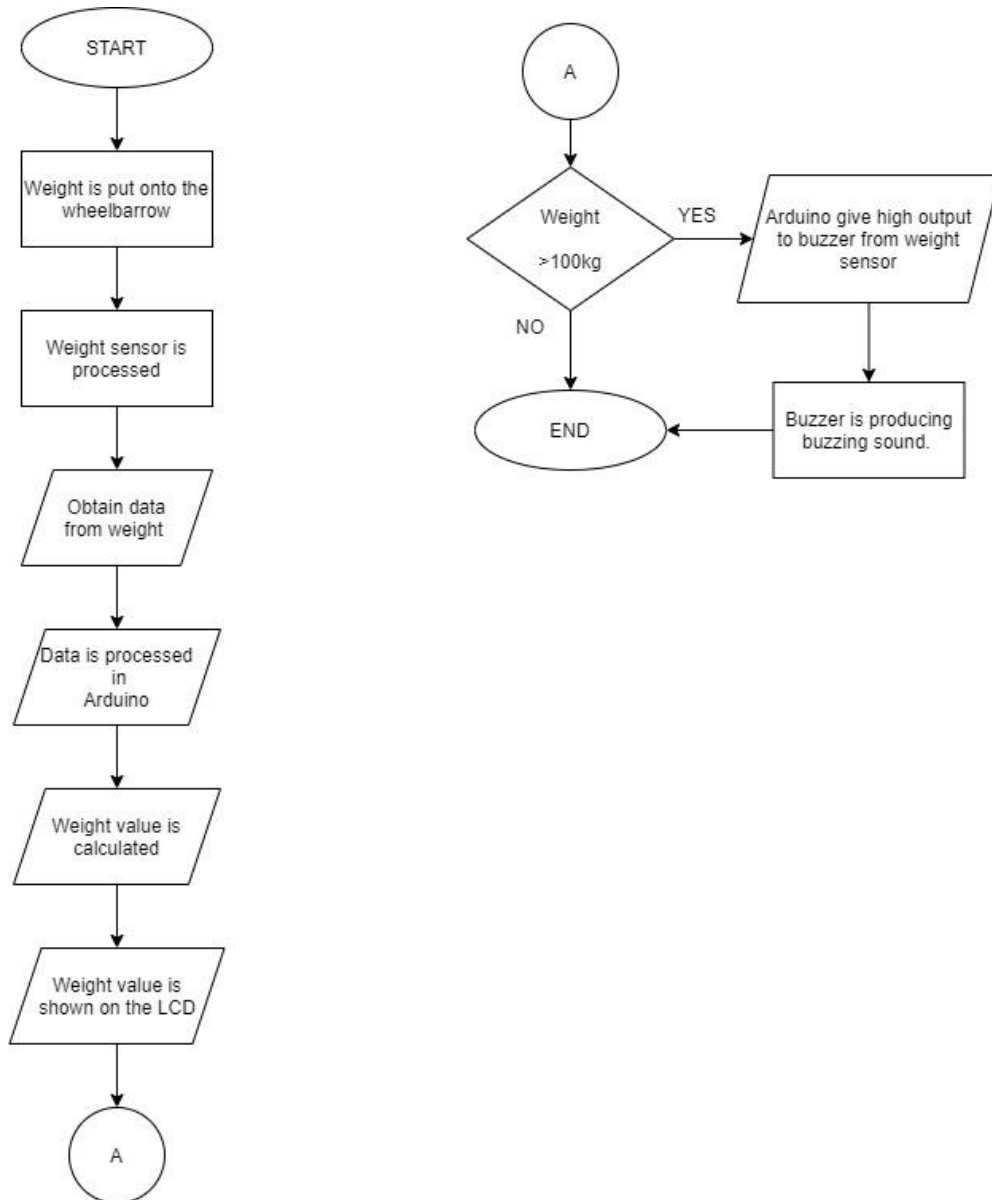
In this project, one of the outputs of the overall system is a linear actuator that raises the barrow up and down without using human energy. When we first get this project, we decide to use Arduino Mega to control both linear actuator and load cell. Our initial system is the Arduino Mega will get input weight from load cell and send the output to the LCD to display the weight. Depend on the weight from weight sensor the Arduino Mega will give permission for the actuator to lift up as safety measure, then user can push button to lift the barrow. Our first coding can be seen from appendix B. But after some thinking, we decide to use two Arduino Uno. The reason why is because we need precaution on something happen to Arduino Mega. If one of the Arduinos Uno for an example one that control the load cell damage, the linear actuator can still function we just need to change the broken part. Beside the price of one Arduino Mega is the same with two Arduino Uno. The new coding can be seen in appendix A and appendix C.

With this, we decided to use the previous system but change the weight limit of load cell to 100kg. From the previous system, the weight sensor is sensor in the system which send signals to the control system, Arduino uno and then display the weight value on the 2x16 LCD. In the process, HX711 also needed to amplify the signal of the weight sensor. Not only that, a PWM motor driver also needed in the system to control the linear actuator forward and reverse. The linear actuator will be connected to PWM motor driver and the PWM motor driver will connected to Arduino Uno. The Arduino Uno will be coded to give command to PWM motor driver to do the automation lift up and down. We also do a safety system to protect the linear actuator. When weight sensor detected the load is more than 100kg, it will deactivate the linear actuator as well as alert user that the load is overweight and not suitable to run. This is used to protect our motorised wheelbarrow not carry load over the limit of max load to ensure our motorised wheelbarrow can last longer.



**Figure 3.2:** Flowchart of The System Run

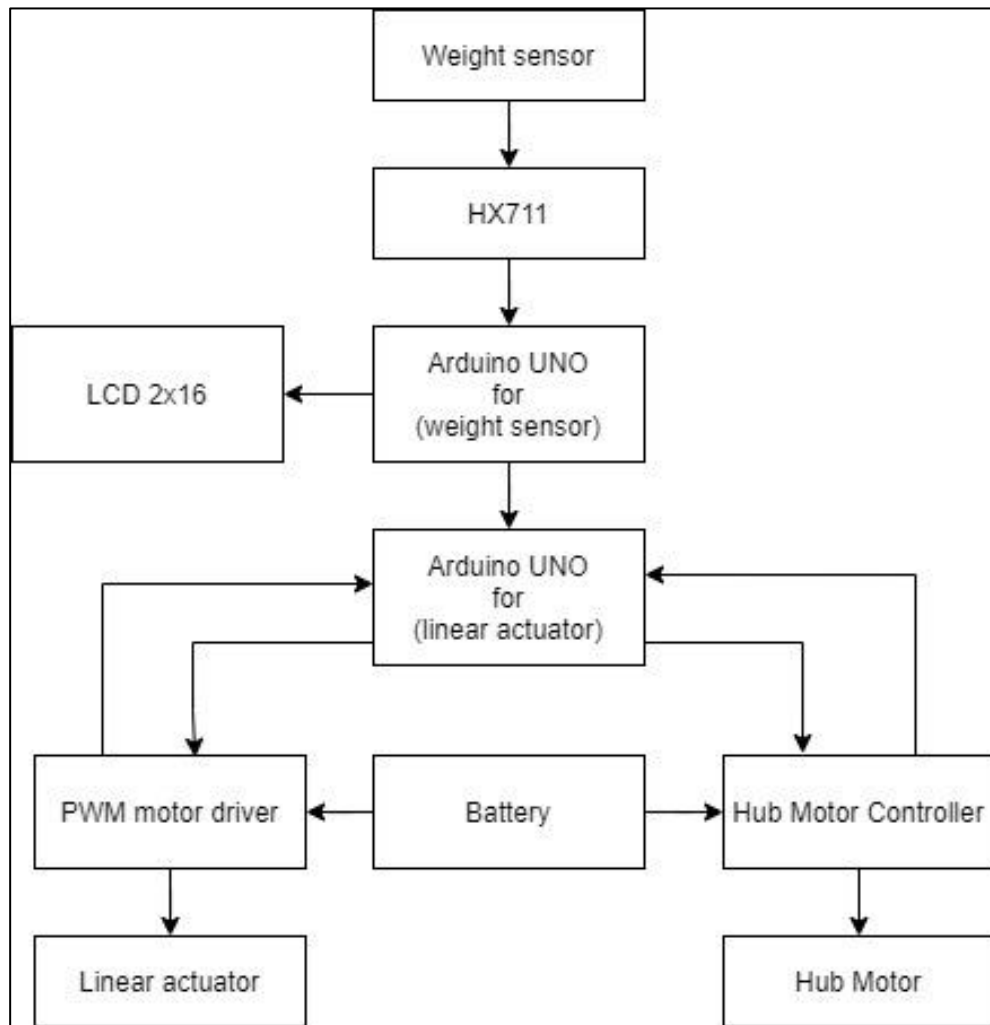
### 3.3 Weight Sensor Operation



**Figure 3.3:** Weight Sensor Operation

### 3.4 Project Flow

This project composed of linear actuator, battery, HX711, weight sensor, hub motor, PWM motor driver, LCD 2x16 and Arduino UNO. We use Arduino software IDE and Proteus 8. Below is a block of the system.



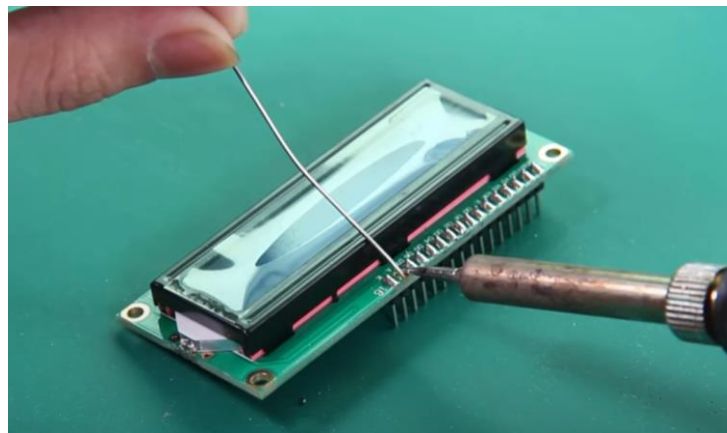
**Figure 3.4:** Motorized Wheelbarrow System Block Diagram



### 3.5 Wiring and Soldering

Wiring and soldering is an important part in this project. This is because there are many connecting parts between different component such as Arduino UNO, LCD, PWM motor driver, and more. Before we solder or do wiring, we need to know the connection between each component. First, we test the connection on the simulation in Proteus 8. After the simulation show no error, we continue with connection on the stripboard. The first thing that we do is interfacing between Arduino UNO, LCD, HX711 and weight sensor. Then we continue with Arduino UNO, PWM motor driver and linear actuator. Lastly, we combine both Arduino and make a safety system where the actuator will be deactivated when there more 100kg load.

During the soldering process, we must know which pin connected to which and solder it carefully. We must be careful to not add to much and to little soldering wire. If the solder is too much, sometime it will connect two joints. Use a solder sucker to help get rid of the excess. After done with the soldering, check the component before move on to solder another part. This can prevent loss of time when something wrong with the soldering.



**Figure 3.5:** Soldering

### 3.6 Material

#### 3.6.1 Battery

For the main source of this project, we use 12V rechargeable lead acid battery. The lead acid batteries use to power Arduino, linear actuator and hub motor. This type of battery can produce enough power to run the system. The table below shows the general condition of the battery when charging and discharging.

Table 1: Battery state of charge

State of Charge	12 Volt battery	Volts per Cell
100%	12.7	2.12
90%	12.5	2.08
80%	12.42	2.07
70%	12.32	2.05
60%	12.20	2.03
50%	12.06	2.01
40%	11.9	1.98
30%	11.75	1.96
20%	11.58	1.93
10%	11.31	1.89
0	10.5	1.75

A typical 12 V lead-acid battery has six cells connected in series, each of which delivers about 2 V. each cell contains two lead grids packed with the electrode material. The grids are immersed in an electrolyte solution. Fiberglass sheets between the grids prevents shorting by accidental physical contact. When the cell discharges, it generates electrical energy as a voltage.

We choose this battery because we like the advantages that it gives. The first one, it has longest life cycle from 250 to 750 cycles. Next, it can be used on high and low temperature. This battery also can be rechargeable. Because of this advantage, we decided this battery is the most suitable.

### 3.6.2 Linear Actuator

This linear actuator pushes and pulls a rod inside the case. Inside the bottom portion of the linear actuator, there are several gears and magnets for pushing the rod and pulling the rod. The linear actuator has only positive and negative wire, this makes it easier for us to change the direction of the rod by changes the polarity of the wire. For this project, the linear actuator is use as the mechanism to pulled and pushed barrow. This linear actuator is placed bellow the barrow near the back of it.



**Figure 3.6:** Heavy Duty Linear Actuator

Features of Linear actuator: -

Material: Aluminium alloy

Rated power: 20W, maximum 30W

Load capacity: 1500N

Input Voltage: 12 V Dc

Speed: 1250mm/s

Stroke length: 200mm/8inch

Working frequency: 20%

### 3.6.3 MD10C R3 Motor Driver

For this project, MD10C R3 motor driver was used to control the behaviour of linear actuator. It come with a very high current support up to 10A. for this type of motor driver, it can control one dc motor. The specialty of this motor driver, it has a manual button for testing. The manual button can control the direction of the linear actuator motor. It is easy to use with a little guide. The figure below shows the label for connection, indicator and the dimension.

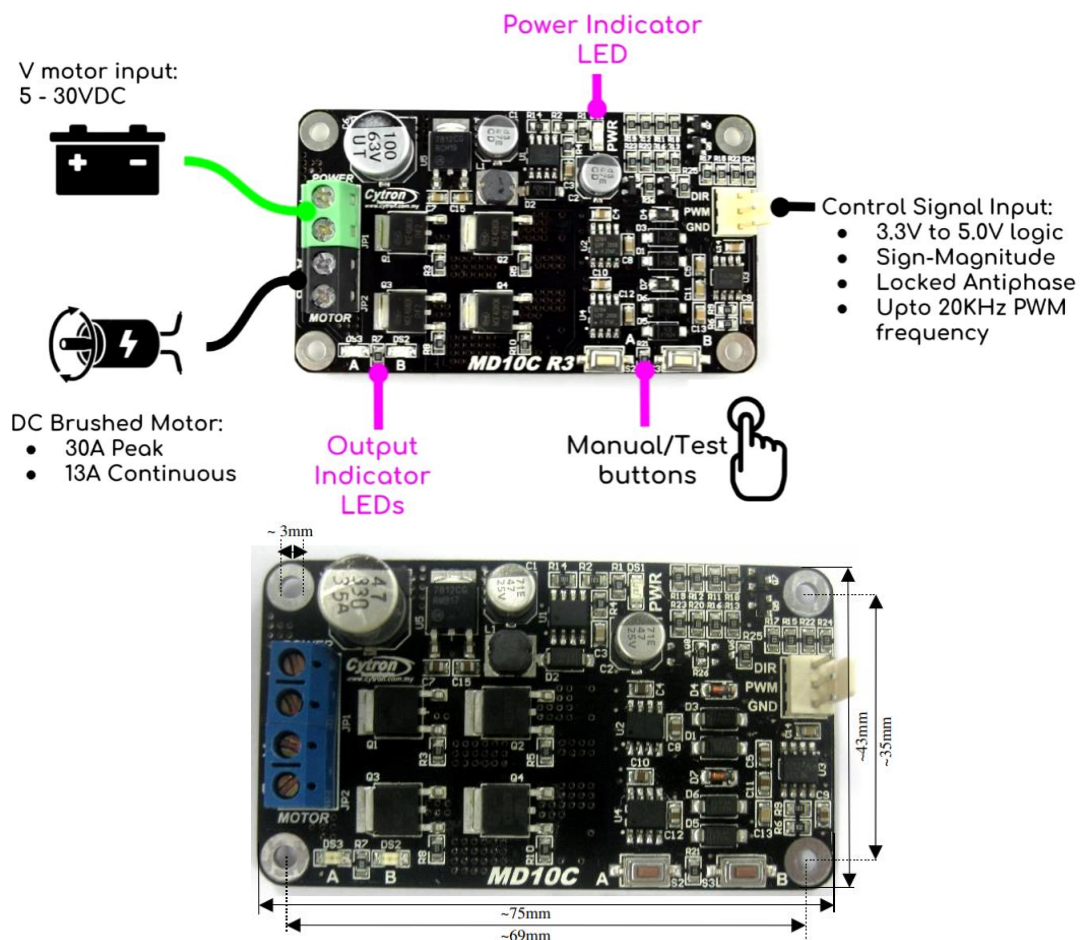


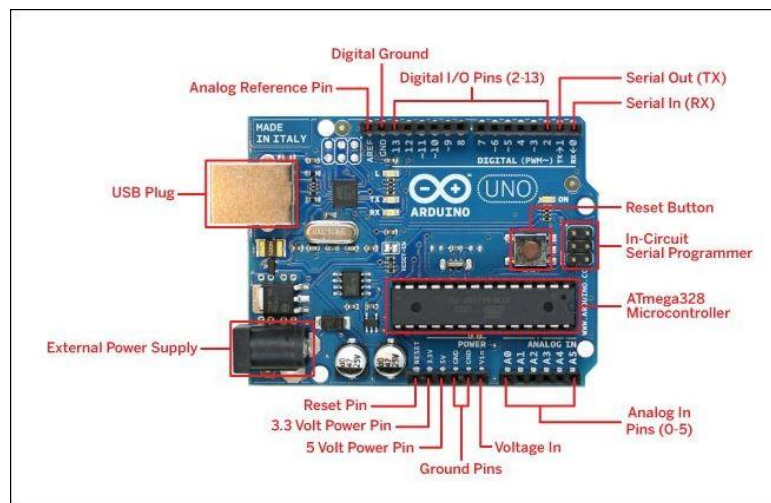
Figure 3.7: MD10C R3 PWM Motor Driver

Specification/Features:

- Bi-directional control for one Brushed DC motor (Single Channel).
- Support motor voltage ranges from 5V to 30VDC.
- No Reverse Polarity Protection at V motor
- Regenerative Braking.
- Maximum current up to 13A continuous and 30A peak (10 seconds).
- 3.3V and 5V logic level input, compatible with Arduino and Raspberry Pi.
- Solid state components provide faster response time and eliminate the wear and tear of mechanical relay.
- Speed control PWM frequency up to 20KHz (output frequency is same as input frequency).

### 3.6.4 Arduino UNO

For this project, we use Arduino UNO R3 as the microcontroller. We use two Arduino UNO in this project. The first one is to check the weight from weight sensor is it safe or overweight then it sends the output to second Arduino UNO to move the linear actuator or not depend on the signal it gives. We use this method as safety measure for our project. If one of the Arduinos is damage, we can replace the Arduino, because it really cheap. Besides, two Arduino UNO is the same price as one Arduino Mega in the market so it can save money.



**Figure 3.8:** Arduino UNO R3

Specification: -

Operating Voltage: 5V

Input Voltage: 7-12V

Length: 68.6 mm

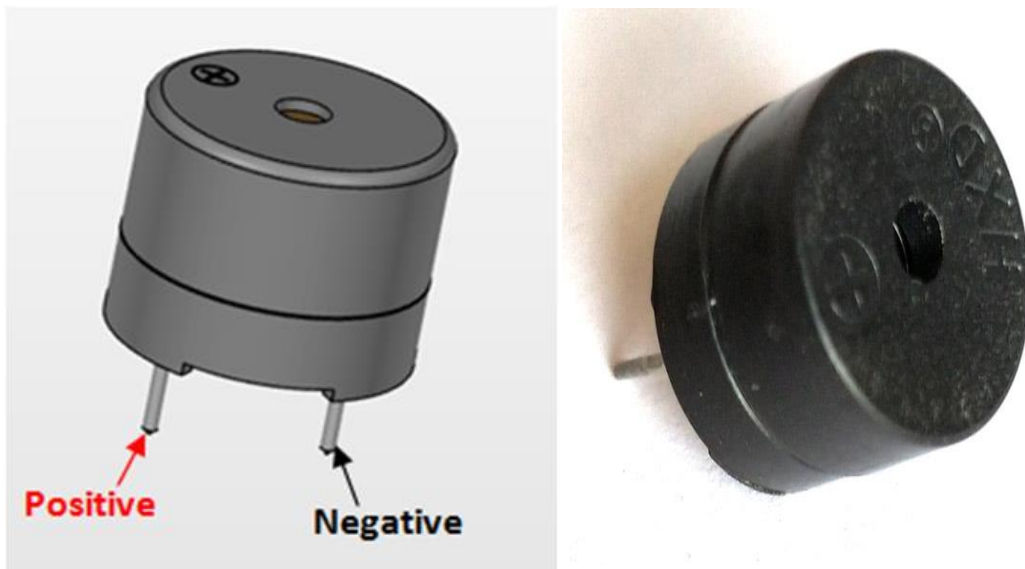
Width: 53.4 mm

Weight: 25 g

### 3.6.5 Buzzer

In this project, we need some kind of warning to inform the user when the load reaches the maximum capacity. So, the user know they will need to remove some weight in the wheelbarrow.

A buzzer is perfect for this job as an audio signal device to inform the user. It is small in size hence it can be easily place in small area. The buzzer can be used by supplying DC power supply ranging from 4 V to 9 V. it is recommended to use a regulated +5V DC supply.



**Figure 3.9:** Buzzer

Specifications: -

- Colour : Black
- Buzzer Type : Piezoelectric
- Sound Level : 95 dB
- Rate Voltage : 12V DC
- Operating Voltage : 3 - 24V
- Max Current Rating: 10mA
- Frequency :  $3900 \pm 500$ Hz

### **3.6.6 Weight Sensor Strain Gauge**

In order to measure the weight, the strain gauge was chosen. It is a sensor whose resistance varies with applied force. Stress and strain are the outcome if external forces are applied to a stationary object. Typical strain gauge resistances range from  $30 \Omega$  to  $3k \Omega$  (unstressed). The sensor will be connected to HX711 and Arduino UNO. Four weight sensor strain gauges will be used and bonded together to make it more sensitive. Each strain gauge can measure up to 50kg. The sensor will become a Wheatstone Bridge circuit when four sensors combine and can measure up to 200kg.





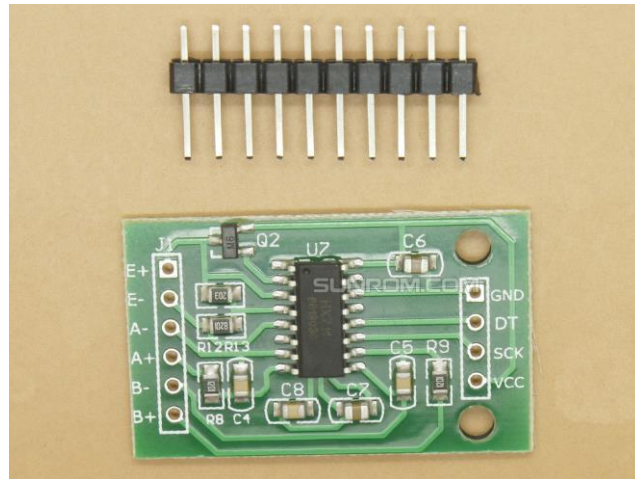
**Figure 3.10:** Weight Sensor Strain Gauge

Specification: -

- Capacity: 50kg
- Total Size: 34 x 34 x 3mm
- Cable Length: 40cm
- Material: Aluminium Alloy
- Weight: 18g

### 3.6.7 HX711

HX711 is a 24-bit Analog-to-Digital converter (ADC) designed for weigh scale or interface directly with a bridge sensor. By connecting it to microcontroller, it will be able to read the changes in the load cells to get accurate weight measurements with some calibration. HX711 connected to Wheatstone Bridge to change the analog from sensor and turn it to digital and send it to Arduino to be calibrate.



**Figure 3.11:** HX711

Specification: -

- Differential input voltage:  $\pm 40\text{mV}$  (Full-scale differential input voltage is  $\pm 40\text{mV}$ )
- Data accuracy: 24 bit (24 bit A / D converter chip.)
- Refresh frequency: 10/80 Hz
- Operating Voltage: 2.7V to 5VDC
- Operating current:  $<10\text{ mA}$
- Size: 24x16mm

## CHAPTER 4

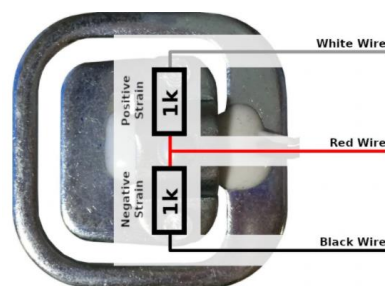
### RESULTS AND DISCUSSION

#### 4.1 Introduction

This chapter will show a step-by-step overview of how to implement the motorized wheelbarrow. The main focus in this chapter is wiring and interfacing, calibrating and coding. The method will be explained in this chapter.

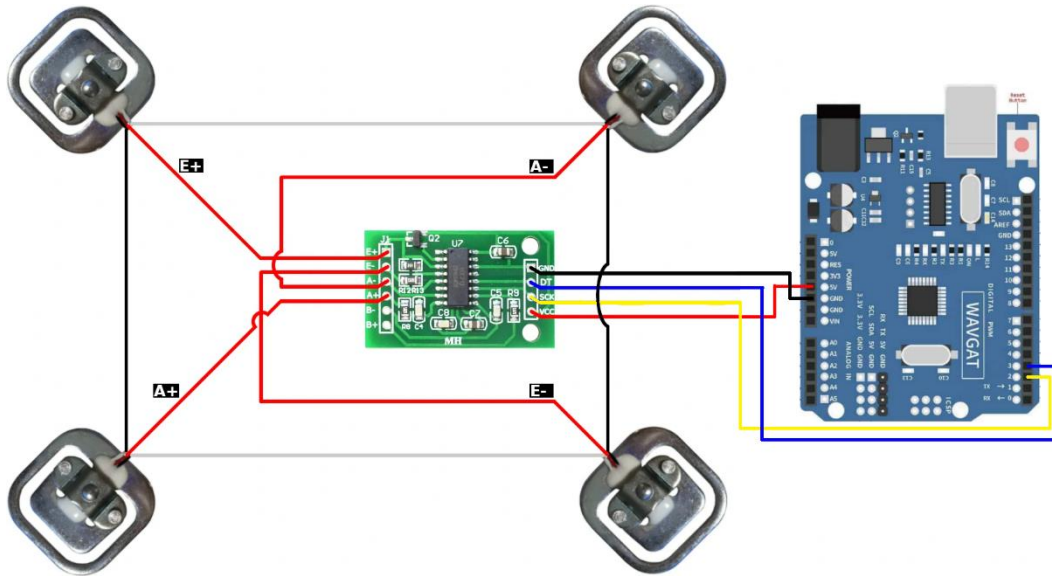
#### 4.2 Weight Sensor Strain Gauge Connection

For the connection of the weight sensor, we connect it according to Wheatstone Bridge. The reason we used Wheatstone Bridge is because the ability to provide a very accurate measurement.



**Figure 4.1:** Weight sensor

When putting the load on the weight sensor, it creates positive and negative strain as shown in figure 4.1. The white line represents positive strain, while the black line represents negative strain. When all the weight sensor has been assembled with HX711, it does not need any resistor. Due to the shape of the Wheatstone Bridge circuit, the diagonally-opposite sides are each working together, as is shown in figure 4.2.



**Figure 4.2:** Setup of weight sensor

To assemble the weight sensor, form a loop by linking the same colour outer wires (white and black of the four sensors. Confirm the resistance of both diagonals (red wires). Both of them should be around 2k ohms. Next, connect the middle cable (red) of one of the diagonals to the E+ and E- outputs of the HX711 module. E+ and E- are the power wires for the cells. The polarity doesn't matter. Switching will only invert the calibration parameter in the software. After that, connect the other diagonal to the A+ and A- inputs of the HX711 module. A+ and A- are the measurement inputs from the cells. Like with the power wires, the polarity is not important. Besides, connect the GND of the HX711 module to the Arduino GND and VCC to the Arduino 5V pin. HX711 also works with 3.3V. So, if you have some other microcontroller that runs on 3.3V, then you can use 3.3V instead of 5V. Lastly, Connect the DT and SCK of the HX711 module to any of the Arduino digital I/O pins.

### 4.3 Calibration

We need to convert the signal that we read to a value we can work with before weight sensor calibration begins. We used the library HX711-ADC that can be found on GitHub. We can read values from weight sensors now, but it will not be going to make too much sense. Secondly, we remove all weight from the scale, connect the Arduino, and power it up. Include the HX711 library, create a new scale and call *set\_scale()* and *tare()* in the *setup()-method*.

```
void setup()
{
  HX711 scale(DOUT, CLK);

  scale.set_scale();

  scale.tare();
}
```

*tare()* resets the current weight to zero. This can be useful in case we want to weigh the contents of a container, without weighing the container itself. However, as we removed all weight before, the scale should be set to zero now. Now we get to the actual calibration. Create a variable called *calibration\_factor*, and set it to zero, or a value of your choice.

```
float calibration_factor = 0;
```

Upload the code to the Arduino. Once the *loop()-function* has been reached (the printed weight should be zero), place an object with a known weight on the scale. The scale will now likely report something completely different from what you want, but this is where the *calibration\_factor* plays its role. Simply send + to increase or – to decrease the calibration factor to the Arduino. Repeat this until it reports the weight you’re looking for.

Finding the correct value might take some time, but needs to be done only once. We can now use the scale by setting the calibration-factor you found in the setup()-method with

```
scale.set_scale(calibration_factor);
```

In the loop()-method, call

```
float weight = scale.get_units();
```

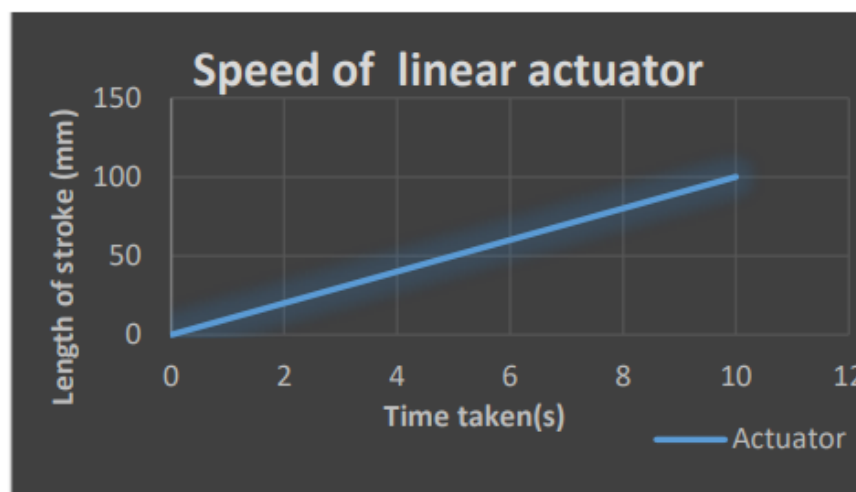
and we can finally measure whatever we built the scale for. The done coding is put at the Appendix C.



**Figure 4.3:** Calibrating weight

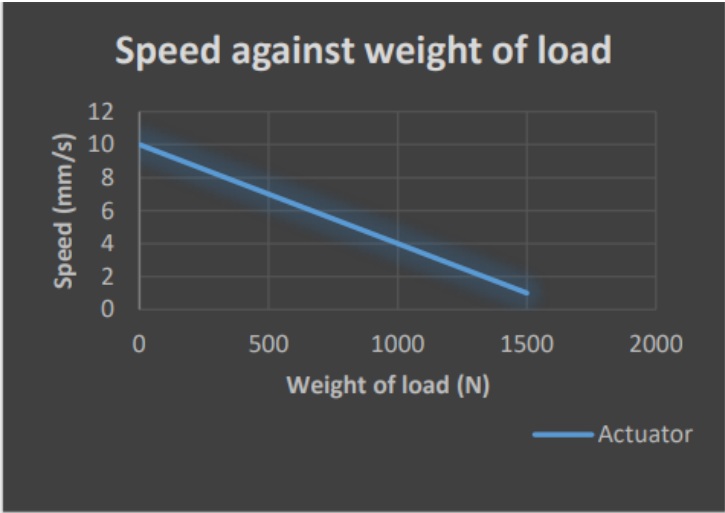
#### 4.4 Linear Actuator

First of all, the improvement is the position of linear actuator. The linear actuator is fixed below the bucket that can push directly and look great. We also using high speed and short length of linear actuator that can push faster and reduced unwanted space. Secondly, the improvement is the position and size of the switch box. The switch box is minimized and put at the bottom of motorized wheelbarrow. The switch buttons are put at the handle that can push the button easily.



**Figure 4.4:** Speed of linear actuator

Based on the result in figure 4.4, it shows that the speed of linear actuator when push the button momentarily. These specifications of linear actuator are 1500N load capacity, 10mm/s speeds and 200mm stroke length. This linear actuator is more suitable compare to other with lower speeds.



**Figure 4.5:** Speed against weight of load

Based on result in figure 4.5, it shows the speed against weight of load. The result show that when the weight of load increases, the speed of linear actuator will decrease. It can lift up or down 150kg but with lower the speed.



## **4.5 Wiring Connection**

To avoid the incorrect relation between the components, we need to link all the components together so that it will work properly. It is necessary to carefully design the circuit so that the weight sensor can detect the well-functioning load and linear actuator, and this was accomplished by the circuit we designed. In Appendix D and E, the model circuit diagram is shown. Based on the wiring diagram we build on Proteus 8, depending on the design, we begin to attach to all electrical components in the breadboard. Then, we start to connect the electrical parts with PWM motor driver, HX711 and Arduino UNO. We make wiring connections from the Arduino UNO to the LCD and weight sensor.

The PWM motor driver receives the 12V battery input power supply and the linear actuator output. By connecting directly to the Arduino UNO, we control the PWM Motor Driver input button. For doing this, we can code the UNO Arduino to give the PWM Motor Driver button high performance and we just have to briefly push the button and it will automatically unload it. Finally, as seen, we fit all the components well into the PVC box.

## **CHAPTER 5**

### **CONCLUSION**

#### **5.1 Conclusion**

As conclusion, the motorized wheelbarrow was successfully improved a bit more by changing the position and high speed of linear actuator, position change and minimize the switch box. 12 mm/s speed of linear actuator is fixed under the bucket to push the bucket when the button pushes momentarily. The LCD display the weight of load when the weight sensor is detected, the Arduino UNO send a feedback and give an output to the LCD. Then, linear actuator lifts up with weight under 100kg while buzzer on and linear actuator cannot lift up when weight over 100kg. Although, the hub motor cannot be used because it broken, the wheelbarrow still can be used to measure the weight and unload the load.

#### **5.2 Recommendation**

For future reference, there still some demerit that I find during the length of this project. There is still some improvement that can be made to increase the efficiency of this wheelbarrow. Below are some of the recommendation that can be made for future reference of the Motorized Wheelbarrow:

- Replace the damage hub motor.
- Build a better switch station at wheelbarrow handle
- Replace rivet part with something stronger like screw, bolt and nuts.

## REFERENCES

Yemini Z, inventor; Zag Industries Ltd, assignee. Wheelbarrow. United States patent US 6,193,265. 2001 Feb 27.

Hoover TA, inventor; Hoover, Terry A., assignee. Motorized wheelbarrow. United States patent US 5,465,801. 1995 Nov 14.

Etzioni Y, Solomon P, Glazman M, inventors; Tzora Active Systems Ltd, assignee. Hub motor. United States patent US 7,249,643. 2007 Jul 31.

"Linear actuator." U.S. Patent 6,158,295, issued December 12, 2000.

M.Sreenivasulu, P.Vijaya Bhaskara Reddy, International Journal of Applied Engineering Research ISSN 0973-4562 Volume 14, Number 4 (2019) pp. 930-935

Saurabh Kadia, Journal Improvisation and fabrication of HybridMotorcycle, IJARIE-ISSN(O)-2395-4396, Vol-4 Issue-5 2018

Pooja Iyer M, G Ravi Teja, V Sitaram Prasad, Journal Design and Fabrication of Solar Electric Scooter, ISSN (Print): 2320-9356, Volume 2 Issue 5 | May. 2014 | PP.21-28

Uno A. Arduino Uno. online), <https://store.arduino.cc/usa/arduino-uno-rev3>, diakses. 2019 Jul;4.

Corcoran, Kieran M. "When Does the Buzzer Sound: The Nonstatutory Labor Exemption in Professional Sports." Colum. L. Rev. 94 (1994): 1045.

Avia semiconductor, HX711 24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales technical report.

## **APPENDICES**

## APPENDIX A

### **CODING ADRUINO PART LINEAR ACTUATOR**

```
const int inputmotorFWDPin =A0;

const int inputmotorREVPin =A1;

const int PWMPin =A2; // from adruino UNO

const int outputmotorFWDPin =4;

const int outputmotorREVPin =6;

const int overweightPin= 7;

const int motorspeed=A4;// control speed

void setup() {

Serial.begin(9600);// read serial monitor

pinMode(inputmotorFWDPin,INPUT); //input forward button

pinMode(inputmotorREVPin,INPUT); // input reverse button

digitalWrite(inputmotorFWDPin,HIGH); // give high output for normally open button

digitalWrite(inputmotorREVPin,HIGH);

pinMode(outputmotorFWDPin,OUTPUT); //output Forward motor

pinMode(outputmotorREVPin,OUTPUT); // output Reverse motor

pinMode(PWMPin,input);// read PWM signal from adruino weight sensor

pinMode(motorspeed,output);

pinMode (overweightPin,input);}

void loop() {

int input1=digitalRead(inputmotorFWDPin);

int input2=digitalRead(inputmotorREVPin);

Serial.println(input1); // monitor forward button status

delay(1);

Serial.println(input2); // monitor reverse button status
```

```

delay(1);

int PWM =analogRead(PWMPin);

int overweight= digitalRead(overweightPin);

// put your main code here, to run repeatedly:

if (PWM== 54)//very light
{
if (input1==LOW&&input2==HIGH&&overweight==LOW) // when forward button is
push
{ analogWrite(motorspeed,54)

digitalWrite(outputmotorFWDPin,LOW);

delay(60000); //linear actuator rise for 1 minute

digitalWrite(outputmotorREVPin,HIGH);

delay(1000);//delay for 1 second to avoid crash.

digitalWrite(outputmotorREVPin,LOW);

delay(60000);// linear actuator revert back for 1 minute

}

else if (input1==HIGH&&input2==LOW&&overweight==LOW) //when reversed button is
pushed

{ analogWrite(motorspeed,54)

digitalWrite(outputmotorFWDPin,HIGH);

digitalWrite(outputmotorREVPin,HIGH);

delay(1000);

digitalWrite(outputmotorREVPin,LOW);

delay(60000);// linear actuator revert back for 1 minute

}

else

{ digitalWrite(outputmotorFWDPin,HIGH);

digitalWrite(outputmotorREVPin,HIGH);}

```

```

}
else if (PWM== 85)//light
{
if (input1==LOW&&input2==HIGH&&overweight==LOW) // when forward button is
push
{ analogWrite(motorspeed,85)
digitalWrite(outputmotorFWDPin,LOW);
delay(60000); //linear actuator rise for 1 minute
digitalWrite(outputmotorREVPin,HIGH);
delay(1000);//delay for 1 second to avoid crash.
digitalWrite(outputmotorREVPin,LOW);
delay(60000);// linear actuator revert back for 1 minute
}
else if (input1==HIGH&&input2==LOW&&overweight==LOW) //when reversed button is
pushed
{ analogWrite(motorspeed,85)
digitalWrite(outputmotorFWDPin,HIGH);
digitalWrite(outputmotorREVPin,HIGH);
delay(1000);
digitalWrite(outputmotorREVPin,LOW);
delay(60000);// linear actuator revert back for 1 minute
}
else
{ digitalWrite(outputmotorFWDPin,HIGH);
digitalWrite(outputmotorREVPin,HIGH);
}
}
}

```

```

else if (PWM== 149)//moderate
{
if (input1==LOW&&input2==HIGH&&overweight==LOW) // when forward button is
push
{ analogWrite(motorspeed,149)
digitalWrite(outputmotorFWDPin,LOW);
delay(60000); //linear actuator rise for 1 minute
digitalWrite(outputmotorREVPin,HIGH);
delay(1000);//delay for 1 second to avoid crash.
digitalWrite(outputmotorREVPin,LOW);
delay(60000);// linear actuator revert back for 1 minute
}
else if (input1==HIGH&&input2==LOW&&overweight==LOW) //when reversed button is
pushed
{ analogWrite(motorspeed,149)
digitalWrite(outputmotorFWDPin,HIGH);
digitalWrite(outputmotorREVPin,HIGH);
delay(1000);
digitalWrite(outputmotorREVPin,LOW);
delay(60000);// linear actuator revert back for 1 minute
}
else
{ digitalWrite(outputmotorFWDPin,HIGH);
digitalWrite(outputmotorREVPin,HIGH);
}
}
else if (PWM== 213)//heavy
{

```



```

if (input1==LOW&&input2==HIGH&&overweight==LOW) // when forward button is
push
{analogWrite(motorspeed,213)
digitalWrite(outputmotorFWDPin,LOW);
delay(60000); //linear actuator rise for 1 minute
digitalWrite(outputmotorREVPin,HIGH);
delay(1000);//delay for 1 second to avoid crash.
digitalWrite(outputmotorREVPin,LOW);
delay(60000);// linear actuator revert back for 1 minute
}
else if (input1==HIGH&&input2==LOW&&overweight==LOW) //when reversed button is
pushed
{ analogWrite(motorspeed,213)
digitalWrite(outputmotorFWDPin,HIGH);
digitalWrite(outputmotorREVPin,HIGH);
delay(1000);
digitalWrite(outputmotorREVPin,LOW);
delay(60000);// linear actuator revert back for 1 minute
}
else
{digitalWrite(outputmotorFWDPin,HIGH);
digitalWrite(outputmotorREVPin,HIGH);
}
}
else if (PWM== 255)//very heavy
{

```

```

if (input1==LOW&&input2==HIGH&&overweight==LOW) // when forward button is
push
{analogWrite(motorspeed,255)
digitalWrite(outputmotorFWDPin,LOW);
delay(60000); //linear actuator rise for 1 minute
digitalWrite(outputmotorREVPin,HIGH);
delay(1000); //delay for 1 second to avoid crash.
digitalWrite(outputmotorREVPin,LOW);
delay(60000); // linear actuator revert back for 1 minute
}
else if (input1==HIGH&&input2==LOW&&overweight==LOW) //when reversed button is
pushed
{ analogWrite(motorspeed,255)
digitalWrite(outputmotorFWDPin,HIGH);
digitalWrite(outputmotorREVPin,HIGH);
delay(1000);
digitalWrite(outputmotorREVPin,LOW);
delay(60000); // linear actuator revert back for 1 minute
}
else
{ digitalWrite(outputmotorFWDPin,HIGH);
digitalWrite(outputmotorREVPin,HIGH);
}
}
}
}

```

## APPENDIX B

### CODING ADRUINO PART WEIGHT SENSOR (PREVIOUS)

```
#include<Servo.h>

#include "HX711.h"

HX711 scale(4,5); //HX711 scale(6, 5);

#include <LiquidCrystal.h>

#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2); // Set the LCD address to 0x27 for a 16 chars
and 2 line display

#define buzzer  8 // buzzer pin

#define ledred  12 // led pin

#define ledgreen  13 // led pin

#define relay  9 // led pin

float calibration_factor = -24;

float units;

float units_1;

Servo servo;

void setup()

{

  Serial.begin(9600);

  pinMode(relay, OUTPUT);

  pinMode(buzzer, OUTPUT);

  pinMode(ledred, OUTPUT);

  pinMode(ledgreen, OUTPUT);
```

```
digitalWrite (relay,HIGH);

Serial.println("HX711 weighing");
scale.set_scale(calibration_factor);
scale.tare();
Serial.println("Readings:");
lcd.init(); // initialize the lcd
lcd.init();
lcd.backlight();

lcd.print("  Weight ");
lcd.setCursor(0,1);
lcd.print(" Measurement ");
delay(5000);
lcd.clear();
}
void loop()
{
  Serial.print("Reading:");
  units = scale.get_units(),10;
  if (units < 0)
  {
    units = 0.00;
  }
}
```

```

units_1 = units /1000;

Serial.print( units_1);

Serial.println(" Kg ");

lcd.setCursor(0,0);

lcd.print("Weight = ");

lcd.setCursor(8,0);

lcd.print(units_1  );

lcd.setCursor(12,0);

lcd.print(" Kg ");

delay(1000);

digitalWrite(ledgreen,HIGH); // and turn off the LED */

if ( units_1 > 100 ){ // if button is pressed

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Weight = ");

lcd.setCursor(8,0);

lcd.print(units_1  );

lcd.setCursor(12,0);

lcd.print(" Kg ");

lcd.setCursor(0,1);

lcd.print("Status:Over  ");

```

```

digitalWrite (relay,LOW);

digitalWrite (buzzer,HIGH);

digitalWrite (ledred,HIGH);

digitalWrite (ledgreen, LOW);

delay(1000);

}

if (units_1 < 100 ){

    lcd.clear();

    lcd.setCursor(0,0);

    lcd.print("Weight = ");

    lcd.setCursor(8,0);

    lcd.print(units_1  );

    lcd.setCursor(12,0);

    lcd.print(" Kg  ");

    lcd.setCursor(0,1);

    lcd.print("Status:Normal  ");

    digitalWrite (relay,HIGH);

    digitalWrite (buzzer,LOW);

    digitalWrite (ledred,LOW);

    digitalWrite (ledgreen,HIGH);

}

}

```

## APPENDIX C

### CODING ADRUINO PART WEIGHT SENSOR (NEW)

```
#include <HX711.h>

#include <LiquidCrystal.h>

LiquidCrystal lcd(13, 12, 11, 10, 9, 8);

// buzzer wiring

const int buzPin = 7;

//linear wiring

// HX711 circuit wiring

const int LOADCELL_DOUT_PIN = 2;

const int LOADCELL_SCK_PIN = 3;

const int PWMPin=A0

HX711 scale;

void setup() {

// read the input on analog pin 0:

pinMode(buzPin,OUTPUT);

pinMode(PWMPin,OUTPUT);

Serial.begin(38400);

Serial.println("HX711 Demo");

lcd.begin(16, 2);

lcd.print("INITIALIZING.....");

Serial.println("Initializing the scale");

// Initialize library with data output pin, clock input pin and gain factor.

// Channel selection is made by passing the appropriate gain:

// - With a gain factor of 64 or 128, channel A is selected

// - With a gain factor of 32, channel B is selected
```

```

Serial.print("get units: \t\t");

Serial.println(scale.get_units(5), 1); // print the average of 5 readings from the ADC minus
tare weight, divided
// by the SCALE parameter set with set_scale

Serial.println("Readings:");

lcd.clear();

lcd.setCursor(0,0);

lcd.print(" Weight ");

lcd.setCursor(0,1);

lcd.print(" Measurement ");

delay(9000);

lcd.clear();

lcd.print(" Waiting.... ");

}

void loop() {

float weight;

Serial.print("one reading:\t");

Serial.print((scale.get_units()+0.2)/-0.4/10, 2);

Serial.print ("KG");

Serial.print("\t| average:\t");

Serial.println(scale.get_units(10), 1);

scale.power_down(); // put the ADC in sleep mode

delay(5000);

scale.power_up();

lcd.setCursor(0,0);

lcd.print("Measured Weight");

lcd.setCursor(0,1);

```



```

weight=(scale.get_units()+0.2)/-0.4/10;

lcd.print(weight, 2);

lcd.print("Kg ");

if (weight >0&&weight <=10)

{ analogWrite(PWMPin,54);

digitalWrite(buzPin,LOW);

}

else if (weight >10&&weight <=20)

{ analogWrite(PWMPin,85);

digitalWrite(buzPin,LOW);

}

else if (weight >20&&weight <=30)

{ analogWrite(PWMPin,149);

digitalWrite(buzPin,LOW);}

else if (weight >30&&weight <=40)

{

analogWrite(PWMPin,213);

digitalWrite(buzPin,LOW);

}

else if (weight >40&&weight <=50)

{

analogWrite(PWMPin,255);

digitalWrite(buzPin,LOW);}

else if (weight>50)

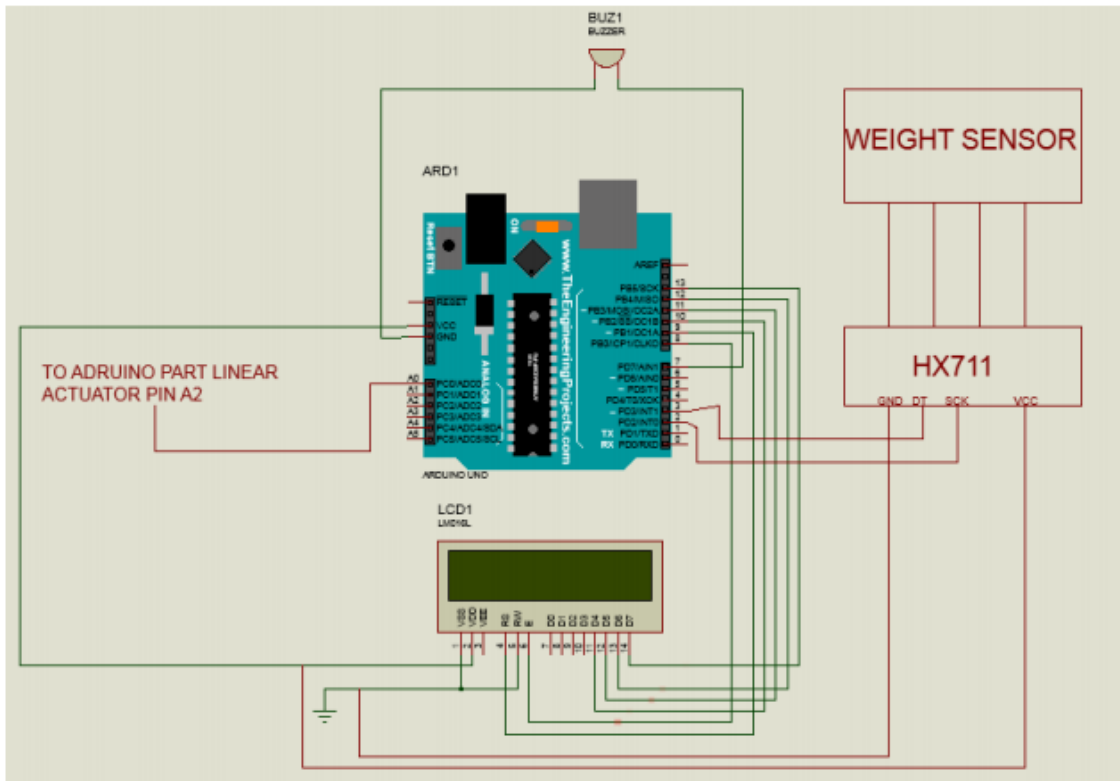
{

lcd.clear();

```

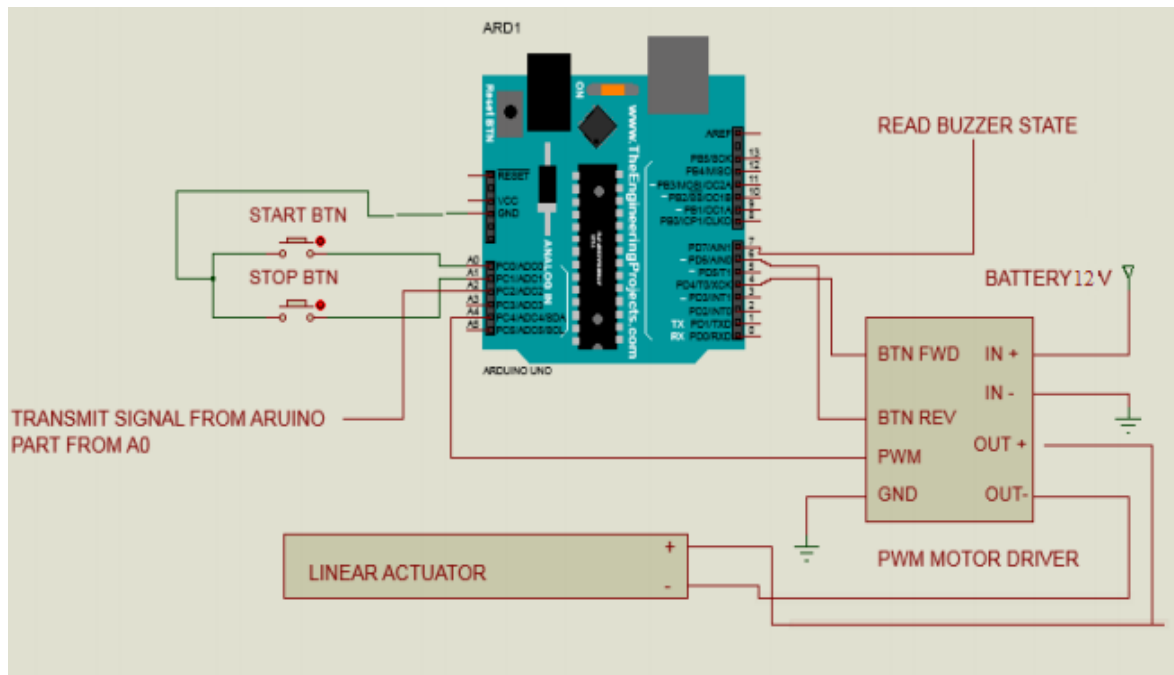
```
lcd.setCursor(0,0);  
  
lcd.print("ATTENTION!!!");  
  
lcd.setCursor(0,1);  
  
lcd.print("OVERWEIGHT!!! ");  
  
delay(5000);  
  
lcd.clear();  
  
digitalWrite(buzPin,HIGH);  
  
}  
  
else  
  
{ digitalWrite(buzPin,LOW);  
  
}  
  
}
```

## APPENDIX D



Circuit Diagram Arduino UNO, LCD, HX711, Weight Sensor

## APPENDIX E



Circuit Diagram Linear Actuator, PWM Motor Driver and Arduino UNO

## APPENDIX F

### Material Cost

No	Item Description	Unit Price (RM)	Quantity	Total Amount (RM)
1	LCD (16x2) Yellow Backlight Model: DS-LCD-162A	RM4.50	1	RM4.50
2	Straight Pin Header (Male) 1x40 Ways Model: CN-PH-M140S	RM1.20	1	RM1.20
3	10Amp 5V-30V DC Motor Driver Model: MD10C	RM46.00	1	RM46.00
4	Donut Board (Small) 6x15cm Model: BDDDB-PP-0615	RM1.60	2	RM3.20
5	3-5V 22x4.5 Piezo Buzzer with Wires Model: SO-PIEZO-22045-W	RM1.00	2	RM2.00
6	40 Ways Male to Male Jumper Wire Model: WR-JW-40MM	RM2.50	1	RM2.50
7	Compatible Arduino Uno Form Factor with USB Cable UNO-CH340	RM17.33	2	RM 34.66
8	50mm-400mm DC 12V 12mm/s IP54 Linear Actuator Motor 1100N Aluminium alloy with 2 Brackets [size: 200mm]	RM142.58	1	RM142.58
9	1500 10mm/s 12V DC Linear Actuator Motor Linear Actuator Lift Electric Motor 200mm	RM106.10	1	RM106.10
10	12 V Battery	RM50.00	1	RM50.00
TOTAL PRICE				RM392.74

## APPENDIX G

### Gantt Chart SDP 1

	FEBRUARY 2020				MARCH 2020				JUNE 2020				JULY 2020			
	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4
SDP 1 Briefing																
Project Meeting																
Gather information																
Project Statement & Design Feature																
Evaluate Design Concept																
Cost Analysis																
First Draft of Proposal																
Second Draft of Proposal																
Proposal Completion																
Preparation of Slide																
SDP 1 Slide Presentation																
Purchase of item																

## APPENDIX H

### Gantt Chart SDP 2

	OKTOBER 2020				NOVEMBER 2020				DECEMBER 2020				JANUARY 2021			
	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4
SDP 2 Briefing																
Project Meeting																
Build up the Design																
Testing the Design with Coding																
Tabulation of Data																
Writing Thesis																
Presentation Rehearsal																
SDP 2 Presentation Day																
Thesis Submission																