

**OEE IMPLEMENTATION ON  
SAFETY HELMET PLASTIC INJECTION PROCESS**

**CHUA YUN CHIN**


**A project report submitted in partial fulfilment of the  
requirements for the award of Bachelor of Engineering  
(Honours) Mechatronics Engineering**

**Lee Kong Chian Faculty of Engineering and Science  
Universiti Tunku Abdul Rahman**

**April 2020**

**DECLARATION**

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature :  \_\_\_\_\_

Name : Chua Yun Chin \_\_\_\_\_

ID No. : 1705116 \_\_\_\_\_

Date : 15/4/2021 \_\_\_\_\_

**APPROVAL FOR SUBMISSION**

I certify that this project report entitled “**OEE IMPLEMENTATION ON SAFETY HELMET PLASTIC INJECTION PROCESS**” was prepared by **CHUA YUN CHIN** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Engineering (Honours) Mechatronics Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature :



Supervisor :

Ir Dr Tan Chee Fai

Date :

15/4/2021

Signature :



Co-Supervisor :

Ir Dr Chuah Yea Dat

Date :

15/4/2021

The copyright of this report belongs to the author under the terms of the copyright Act 1987 as qualified by Intellectual Property Policy of Universiti Tunku Abdul Rahman. Due acknowledgement shall always be made of the use of any material contained in, or derived from, this report.

© 2021, Chua Yun Chin. All right reserved.

## ACKNOWLEDGEMENTS

I would like to thank everyone who had contributed to the successful completion of this project. Firstly, I would like to express my gratitude towards my research supervisor, Ir Dr Tan Chee Fai as well as co-supervisor, Ir Dr Chuah Yea Dat for their professional guidance, advices and patience throughout the project.

In addition, I would like to thank Mr Eugene Lee Sze Kin, General Manager of Proguard Technologies (M) Sdn Bhd for the opportunity given to conduct this research at the company.

Last but not least, I would like to thank my family members who had supported me financially and mentally throughout the entire project.

## ABSTRACT

**“If you can’t measure it, then you can’t improve it.”** (Lord Kelvin, 1907)

Quote above implies how significant this research is in the digital transformation of industry along with Internet of Things (IoT). Data collection is the first and foremost step to achieve continuous improvement (kaizhen) with the utilization of data analytics. This research focused on the implementation of Overall Equipment Effectiveness (OEE) in safety helmet plastic injection process at Proguard Technologies (M) Sdn Bhd. In this paper, OEE computation will be demonstrated on real cases by using Microsoft Excel. Besides, it will reveal the methodology of developing an automatic data acquisition system with IoT capability to collect the data required to perform OEE analysis. By implementing the solution proposed in this paper, Proguard Technologies (M) Sdn Bhd is expected to reduce at least 80% time spent for data collection process. Not only that, the proposed solution also offers many additional features such as cloud monitoring, data analytics, self-triggered call alert, auto-generated email as well as live OEE dashboard.

## TABLE OF CONTENTS

<b>DECLARATION</b>	<b>i</b>
<b>APPROVAL FOR SUBMISSION</b>	<b>ii</b>
<b>ACKNOWLEDGEMENTS</b>	<b>iv</b>
<b>ABSTRACT</b>	<b>v</b>
<b>TABLE OF CONTENTS</b>	<b>vi</b>
<b>LIST OF TABLES</b>	<b>ix</b>
<b>LIST OF FIGURES</b>	<b>x</b>
<b>LIST OF SYMBOLS / ABBREVIATIONS</b>	<b>xiv</b>
<b>LIST OF APPENDICES</b>	<b>xv</b>

### CHAPTER

<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
	1.1 Introduction to Overall Equipment Effectiveness (OEE)	1
	1.2 Importance of the Study	1
	1.3 Problem Statement	2
	1.4 Aims and Objectives	3
	1.5 Scope and Limitation of the Study	3
	1.6 Outline of the Report	3
<b>2</b>	<b>LITERATURE REVIEW</b>	<b>5</b>
	2.1 OEE Factors and its Calculations	5
	2.1.1 Six Major Losses in Production Cycle	6
	2.2 Total Productive Maintenance (TPM)	8
	2.2.1 5S as the Foundation of TPM	8
	2.2.2 Eight Pillars of TPM	10
	2.2.3 Correlation between OEE and TPM	12
	2.3 OEE Implementation	12
	2.4 Plastic Injection Process	17

<b>3</b>	<b>METHODOLOGY AND WORK PLAN</b>	<b>18</b>
3.1	Project Flow	18
3.2	Customize OEE Template	22
3.3	Conceptual Design	22
3.4	Mechanical Design	24
	3.4.1 Conveyor Belt System	25
	3.4.2 Weight Inspection and Rejection Mechanism	27
	3.4.3 Control Box	28
3.5	Electrical and Electronics	30
	3.5.1 ESP32 Microcontroller	31
	3.5.2 HC-SR04 Ultrasonic Sensor	32
	3.5.3 5kg Load Cell with HX711 Amplifier	34
	3.5.4 Momentary Push Button	35
	3.5.5 Active Buzzer	35
	3.5.6 16 x 2 Liquid Crystal Display with I2C Module	36
	3.5.7 28BYJ-48 Stepper Motor with ULN2003 Driver Board	36
3.6	Software Development	37
	3.6.1 Main Program in ESP32 Microcontroller	38
	3.6.2 OEE Computation Program in IoT Server	44
	3.6.3 Low-code Program in Production Dashboard	46
<b>4</b>	<b>RESULTS AND DISCUSSIONS</b>	<b>48</b>
4.1	Manual Data Collection System	48
4.2	Fabrication of Prototype	54
	4.2.1 Mechanical Fabrication	56
	4.2.2 Electrical Connection	57
4.3	Review of Proposed Solution	58
	4.3.1 Cloud Monitoring and Data Analytics	59
	4.3.2 Self-triggered Call Alert	63
	4.3.3 Auto-generated Email of OEE Report	64



4.3.4	Data Monitoring App on Smart Phone	65
4.3.5	OEE Live Dashboard at Production Site	66
4.4	Challenges and Solutions	67
4.4.1	Measurement Noises in Load Cell Sensor	67
4.4.2	Switch Bouncing Issues for Rejection and Downtime Push Button	70
4.4.3	NEMA 17 Stepper Motor Vibration Issue	71
<b>5</b>	<b>CONCLUSION AND RECOMMENDATION</b>	<b>73</b>
5.1	Conclusion	73
5.2	Limitation and Recommendation	73
	<b>REFERENCES</b>	<b>75</b>
	<b>APPENDICES</b>	<b>79</b>

## LIST OF TABLES

Table 1.1:	OEE Performance Score Rankings (Singh, et al., 2018)	1
Table 2.1:	Six Big Losses (Peter, 2018)	6
Table 2.2:	OEE Score Before and After TPM Implementation (Dr. Garg and Gupta, 2012)	13
Table 2.3:	OEE Template for Broaching Machine I	13
Table 2.4:	OEE Calculation Template (Abhishek, et al., 2012)	14
Table 2.5:	Six Big Losses Analysis (Herry, Farida and Lutfia, 2018)	14
Table 2.6:	OEE Structures in India Production Companies (Kumar and Soni, 2014)	15
Table 2.7:	Perspectives of India Production Companies on OEE (Kumar and Soni, 2014)	15
Table 3.1:	BOM of Conveyor Belt System	26
Table 3.2:	BOM of Weight Inspection Module & Rejection Mechanism	28
Table 3.3:	BOM of Control Box	28
Table 3.4:	Specifications of ESP32 Wroom-32 (Espressif Systems, 2014)	32
Table 3.5:	Specifications of HC-SR04 Ultrasonic Sensor (Rui and Sara, 2013)	33
Table 3.6:	Specification of Sparkfun TAL220B Load Cell	34
Table 3.7:	28BYJ-48 Stepper Motor Specifications	37
Table 4.1:	Comparison between Before and After Applying Kalman Filter	68
Table 4.2:	Microstepping Configuration in A4988 Stepper Motor Driver Board	72
Table 5.1:	Comparison between Before and After Applying Proposed Solution	73

## LIST OF FIGURES

Figure 2.1:	Overall Concept of OEE (Amrik, et al., 2010)	6
Figure 2.2:	5S Foundation of TPM	10
Figure 2.3:	Eight Pillars of TPM (Venkatesh, 2005)	12
Figure 2.4:	Data Flow at the Production Site of Motorola (Giegling, et al., 1997)	16
Figure 2.5:	CUBES E10 Template (Konopka, 1996)	16
Figure 2.6:	Injection Moulding Machine	17
Figure 3.1:	Project Flow Chart	19
Figure 3.2:	Gantt Chart of FYP Part I	20
Figure 3.3:	Gantt Chart of FYP Part II	21
Figure 3.4:	Customize OEE Computation Template in Ms Excel	22
Figure 3.5:	Block Diagram of DAQ System	24
Figure 3.6:	Isometric View of The Proposed DAQ System	25
Figure 3.7:	Top View of The Proposed DAQ System	25
Figure 3.8:	Exploded View of Conveyor Belt System	26
Figure 3.9:	Exploded View of Weight Inspection Module & Rejection Mechanism	27
Figure 3.10:	Exploded View of Control Box	29
Figure 3.11:	Graphical Wiring Diagram of DAQ System	30
Figure 3.12:	Schematic Circuit Diagram of DAQ System	31
Figure 3.13:	ESP32 Wroom-32 Pin Layout (Rui and Sara, 2020)	32
Figure 3.14:	Pinout of HC-SR04 Ultrasonic Sensor (Adnan, 2018)	33
Figure 3.15:	5kg Load Cell with HX711 Amplifier (Sarah, 2016)	34
Figure 3.16:	Active Buzzer	35

Figure 3.17:	16 x 2 Liquid Crystal Display	36
Figure 3.18:	28BYJ-48 Stepper Motor with ULN2003 Driver	37
Figure 3.19:	DAQ System Framework	38
Figure 3.20:	Main Program Flowchart	39
Figure 3.21:	Flowchart of External Interrupt for Rejection Push Button	40
Figure 3.22:	Flowchart of External Interrupt for Downtime Push Button	40
Figure 3.23:	Code Segment for Connecting to Wireless Network	41
Figure 3.24:	A Do-while Loop for Counting Total Number of Safety Helmet Produced	42
Figure 3.25:	Millis() Function used in External Interrupts	42
Figure 3.26:	Code Segment to Enable External Interrupts	42
Figure 3.27:	Code Segment for Measuring Weight	43
Figure 3.28:	Code Segment for Checking Load Cell Connection	43
Figure 3.29:	Code Segment for Rejection	43
Figure 3.30:	Code to Activate Buzzer	44
Figure 3.31:	Code Segment for Publishing Collected OEE Data to ThingSpeak	44
Figure 3.32:	Flowchart of OEE Analysis	45
Figure 3.33:	Flowchart of Timer Interrupt	45
Figure 3.34:	Low-code Program in Node-RE	47
Figure 4.1:	Data Collected on 2 <sup>nd</sup> of April 2021 at Proguard Technologies (M) Sdn Bhd	50
Figure 4.2:	OEE Calculations for Line 1	51
Figure 4.3:	OEE Calculations for Line 2	51
Figure 4.4:	OEE Calculations for Line 3	52
Figure 4.5:	OEE Analysis Chart for Line 1	52

Figure 4.6:	OEE Analysis Chart for Line 2	53
Figure 4.7:	OEE Analysis Chart for Line 3	53
Figure 4.8:	OEE Score Indicators	54
Figure 4.9:	Side View of Prototype	54
Figure 4.10:	Top View of Prototype	55
Figure 4.11:	Front View of Prototype	55
Figure 4.12:	Conveyor Belt System	56
Figure 4.13:	Weight Inspection Module with Rejection Mechanism	56
Figure 4.14:	Control Box	57
Figure 4.15:	Electrical Circuit Constructed on Breadboard	57
Figure 4.16:	Electrical Circuit Constructed on Donut Board	58
Figure 4.17:	Available Features in Proposed Solution	59
Figure 4.18:	Graph of Total Number of Units Produced against Time	60
Figure 4.19:	Graph of Rejected Amount against Time	60
Figure 4.20:	Graph of Total Downtime against Time	61
Figure 4.21:	Availability Index	61
Figure 4.22:	Performance Index	62
Figure 4.23:	Quality Index	62
Figure 4.24:	OEE Score	62
Figure 4.25:	Configuration of Self-triggered Call Alert on IFTTT Platform	63
Figure 4.26:	Incoming VoIP Call Alert	64
Figure 4.27:	Configuration of Auto-generated Email on IFTTT Platform	64
Figure 4.28:	Data Monitoring App, ThingShow on Smart Phone	65
Figure 4.29:	Main Page of Live Dashboard	66
Figure 4.30:	OEE Live Dashboard for Line 1	67

Figure 4.31:	Graph of Load Cell Values against Time before Applying Kalman Filter	69
Figure 4.32:	Graph of Load Cell Values against Time after Applying Kalman Filter	69
Figure 4.33:	Switch Bouncing Effect (Christoffersen, 2015)	70
Figure 4.34:	Switch Debouncing Circuit (SujayPosam, 2019)	70
Figure 4.35:	Software Approach to Overcome Switch Bouncing Effect	71

**LIST OF SYMBOLS / ABBREVIATIONS**

BOM	Bill of Materials
CUBES	Capacity Utilization Bottleneck Efficiency System
DAQ	Data Acquisition
FYP	Final Year Project
GPIO	General Purposes Input Output
IoT	Internet of Things
JIPM	Japan Institute of Plant Maintenance
OEE	Overall Equipment Effectiveness
PC	Personal Computer
TPM	Total Productive Maintenance

**LIST OF APPENDICES**

APPENDIX A:	Main Program in ESP32 Microcontroller	79
APPENDIX B:	Matlab Program to Compute OEE Score	93
APPENDIX C:	Matlab Program to Reset Parameters Daily	96



## CHAPTER 1

### INTRODUCTION

#### 1.1 Introduction to Overall Equipment Effectiveness (OEE)

Overall Equipment Effectiveness (OEE) is a performance metric used to evaluate production efficiency in a manufacturing firm. According to Giegling, et al. (1997), OEE was first introduced by a Japanese engineer, Nakajima in 1970. In his book “Introduction to TPM”, Nakajima (1988) suggested that OEE is a practical tool to identify production wastes and reveal hidden losses in daily operation. Liliane and Peter (2010) defined OEE as a quantitative measure used to evaluate the performance of a set of equipment. Other than being a performance indicator, OEE also acts as a driver to boost the performance and efficiency of an organization (Andersson and Bellgran, 2011). According to Singh, et al. (2018), OEE performance score of a manufacturing firm can be divided into four rankings, as shown in Table 1.1 below. OEE of a production company is examined based on three main factors, which are **Availability**, **Performance** as well as **Quality**. Keegan (n.d.) claimed that in recent years, OEE has gained popularity in the manufacturing industry due to its simplicity and accuracy.

Table 1.1: OEE Performance Score Rankings (Singh, et al., 2018)

OEE Score	Level of Performance
100%	Perfect production
85%	World-class manufacturers
60%	Typical manufacturers
40%	Underperforming manufacturers

#### 1.2 Importance of the Study

**“If you can’t measure it, then you can’t improve it.”** (Lord Kelvin, 1907)

This quote implies the significance of OEE in playing a role to improve the production efficiency. Through a better understanding of OEE concepts,

production managers can have a greater control on the capacity utilization of each manufacturing line. Besides that, OEE calculation can identify the underlying losses in daily production. The final outcome of this study may provide a basic guideline to any manufacturing firm that desires to understand or implement OEE.

In this study, the literature of OEE was reviewed and related to TPM. Information about the implementation of OEE in different fields of industry was gathered in Chapter 2 to provide references for further study in this project.

### **1.3 Problem Statement**

Competition in the manufacturing industry today is getting more intense. This is because all manufacturers are striving to minimize their production costs while maintaining product's quality. Fleischer, et al. (2006) claimed that the productivity and availability of production equipment can directly reflect how competitive a manufacturing firm is. Therefore, equipment performance has now become one of the important measures to evaluate the capability of a production company.

Over the years, there are many kinds of research performed to review OEE as a performance indicator for TPM implementation. Unfortunately, it is observed that most of the researches emphasized only at OEE concept and its calculations. They rarely exposed how to acquire OEE data at the production line and send them to information centre for further analysis.

This project is under collaboration with Proguard Technologies (M) Sdn Bhd. Therefore, it will be the main subject of discussion in this report. It is observed that Proguard Technologies (M) Sdn Bhd did not implement any terminology to measure production efficiency and identify major losses in their daily operation. Manual data collection system adopted by Proguard Technologies (M) Sdn Bhd is highly dependent on whiteboard and papers. There are several disadvantages of manual data collection process such as time-consuming, possibility of human errors as well as slow data flow.

#### **1.4 Aims and Objectives**

This project is aimed to implement OEE on safety helmet plastic injection process at Proguard Technologies (M) Sdn Bhd. By using SMART method, objectives defined for this project are:

- (i) To measure machine efficiency at Proguard Technologies (M) Sdn Bhd by performing OEE Analysis via Microsoft Excel.
- (ii) To reduce 80% time spent for data collection process by developing an automatic data acquisition system with IoT capability.

#### **1.5 Scope and Limitation of the Study**

Scope of this study covers from OEE concept until its implementation. The first part of this project is to review the concept of OEE and design a Microsoft Excel template to perform OEE analysis. The second part of this project is to develop an automatic data acquisition system with IoT capability to collect and upload real-time OEE data to cloud for remote monitoring as well as data analytics.

This report focuses on the discussion of OEE implementation as a tool to measure equipment efficiency. However, it provides a very limited reference on the implementation of TPM. Elaboration on TPM concept and its maintenance techniques can only be found in Chapter 2 of this report.

#### **1.6 Outline of the Report**

Chapter 1 of this report provides a general introduction to OEE concept. The insights of this project such as importance of the study, problem statement as well as objectives are clearly defined in this chapter. Scope and limitation of the study are also written to ease the reference and further improvement in future.

Chapter 2 summarizes the literature review of OEE from journal papers and articles. The implementation of OEE in different fields of industry are briefly described in this chapter to provide a basic guideline in implementing OEE on safety helmet plastic injection process.

Chapter 3 explains the methodology used to achieve the objectives of this project. Work plan and Gantt charts for FYP Part I and FYP Part II can be found in this chapter.

Chapter 4 discusses the results and data obtained in this project. Challenges faced and their respective solutions are also documented into this chapter.

Chapter 5 deduces the final outcome of this project and proposes some recommendations for future work.

## CHAPTER 2

### LITERATURE REVIEW

#### 2.1 OEE Factors and its Calculations

As shown in Equation 2.1, OEE score of a production line can be computed by the multiplication of three OEE factors, which are availability, performance as well as quality (Huang et al., 2003). Equation 2.2 shows that availability is the ratio of actual production time against planned production time. It measures the availability of a machine by determining its actual running time over total available time (Jonsson and Lesshammar 1999). Availability losses are often referred to the production events that will contribute to a downtime loss such as equipment breakdown, shortage of raw materials, changeover of staffs, die exchange, initial setup and etc.

As shown in Equation 2.3, performance factor measures the actual output rate of a machine which directly reflects to the equipment's efficiency. All factors that cause the machine to operate at a speed lower than maximum speed are called as performance losses. For example, common performance losses in a manufacturing line are machine jams, machine component wear and inexperienced operators.

Lastly, quality factor expresses the proportion of good products to the total number of units produced in a batch production. It is calculated by using Equation 2.4. Defective products and rejection during production are the typical sources of quality loss. Overall concept of OEE factors is illustrated in Figure 2.1.

$$OEE = Availability \times Performance \times Quality \quad (2.1)$$

$$Availability = \frac{Actual\ Production\ Time}{Planned\ Production\ Time} \quad (2.2)$$

$$Performance = \frac{Total\ Number\ of\ Units\ Produced \times Cycle\ Time}{Actual\ Production\ Time} \quad (2.3)$$

$$Quality = \frac{\text{Number of Good Units}}{\text{Total Number of Units Produced}} \quad (2.4)$$



Figure 2.1: Overall Concept of OEE (Amrik, et al., 2010)

### 2.1.1 Six Major Losses in Production Cycle

In fact, the ultimate target of implementing OEE is to identify and eliminate six big losses in production cycle (Nakajima, 1988). According to Peter (2018), six big losses comprised of unplanned stops, planned stops, minor stops, slow cycles, production rejects as well as start-up rejects. Further explanation for each major loss is shown in Table 2.1.

Table 2.1: Six Big Losses (Peter, 2018)

OEE Factor Category	Type of Loss	Explanation
Availability	Unplanned Stops	Unplanned stops are the production stops due to an unexpected event such as machine failure. Equipment breakdown is usually due to the inappropriate operation by operator and a poor level of maintenance (Ellacott, 2019). Operational errors can be significantly reduced by offering a proper training. At the same time, preventive maintenance

		should be practiced to minimize the sudden failures of machine.
	Planned Stops	Production is scheduled to stop due to a planned activity such as tea breaks, changeover of staff, daily maintenance, machine initial setup and etc. Unlike unplanned stop, planned stop happens every day at production site and it cannot be avoided. However, downtime due to planned stops can be minimized through a proper planning or Kaizen program to improve the process flow.
Performance	Minor Stops	Equipment ceases for a while due to a temporary issue such as material jams, misfeed error, misalignment and etc. It usually refers to a stop that consumes less than 5 minutes.
	Slow Cycles	Equipment is operating at a speed lower than its maximum capability. As a result, it will have a longer cycle time and cause a decrement in the total number of parts produced. Mila (2018) suggested that automation can provide a more reliable cycle time if compared to human operators.
Quality	Production Rejects	It refers to the defective parts created during a mass production. Production rejects can contribute to a significant loss if the rejection

		rate is high and the defective products are non-recyclable.
	Start-up Rejects	It refers to the waste produced during start-up process. It commonly occurs after changeover of die because equipment requires some time and minor adjustments to produce a good quality product and reach a steady state of production.

## 2.2 Total Productive Maintenance (TPM)

Total Productive Maintenance (TPM) is a manufacturing strategy proposed by Nakajima (1988) to improve production efficiency and integrity through the involvement of all employees; from top management level to line operators. It is one of the important elements under the implementation of lean manufacturing. Santiago (n.d.) concludes that TPM is a system that accentuates on the preventive and proactive maintenance to ensure all equipment can operate till their maximum capacities during production. According to Japan Institute of Plant Maintenance (JIPM, 1996), the main objective of TPM is to eliminate and minimize the six major losses in daily production cycle. In order to provide a more comprehensive guideline to industry, JIPM had introduced 5S (Sort, Straighten, Shine, Standardize and Self-discipline) as the foundation of TPM along with eight supporting activities.

### 2.2.1 5S as the Foundation of TPM

5S is a point of commencement for companies that desire to enhance production efficiencies in their workplaces. Michalska and Szewieczek (2007) suggested that adoption of 5S in an organization will not only create a healthy and safety working environment, but also improve the productivity and quality of work done by employee. Figure 2.2 shows the 5S foundation of TPM. According to Maharjan (2011), five techniques proposed in 5S are:



- (i) **Sort (*Seiri*)** – **Classify all things into essential items and non-essential items at manufacturing floor.** It helps workers to determine what is not truly needed at workplace. It is advised that the volume of non-essential items should be always kept as low as possible at workplace.
- (ii) **Straighten (*Seiton*)** – Arrange all things and works in an organized manner. Alvord (2010) had listed down several initiatives to be taken under the category of Seiton:
  - Draw a border line around the machines and equipment
  - Implement an address system in the location of plant to differentiate them effectively
  - Mark and label tools with their respective storage locations
  - Utilize graphics to ease recognition
- (iii) **Shine (*Seiso*)** – Ensure that the tools, machines, equipment and work area are always in a clean condition. It is encouraged to establish a team and plan timetable for regular cleaning routines. Ahuja and Khamba (2008) stated that a clean working environment can not only improve the workers' performance, but also help them in detecting any abnormal circumstances at workplace such as oil leakage, machine malfunction and etc.
- (iv) **Standardize (*Seiketsu*)** – Create standard operating procedures for all important activities. Padilla and Pekmezci (2011) claimed that there are many advantages of implementing standardization in a workplace such as elimination of human errors and wastes due to nescience and wrongful judgement. Standardization can be implemented via optical controls like having a checklist, colour coding as well as labelling.
- (v) **Self-discipline / Sustain (*Shitsuke*)** – Apply 5S program regularly and voluntarily. Management audit should be carried out from time to time to ensure that the established standards are always strictly followed and maintained.

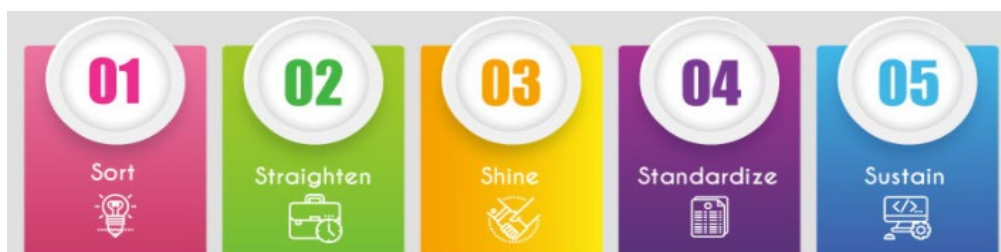


Figure 2.2: 5S Foundation of TPM

### 2.2.2 Eight Pillars of TPM

With 5S as the foundation of TPM, eight supporting activities were introduced to improve the reliability and efficiency of equipment. Figure 2.3 shows the eight pillars of TPM. Eight pillars of TPM are:

- (i) **Focused Improvement (Kobetsu Kaizen)** – Form a team to continuously figure out the possible improvements on the manufacturing flow with the objective of increasing production efficiency. Employees can enhance their problem-solving skills under this pillar.
- (ii) **Autonomous Maintenance (Jishu Hozen)** – Assign the responsibility of taking good care of the equipment to line operators. It can create a sense of ownership in the line operators to protect their own equipment. With a sense of ownership, they are more likely to perform daily maintenance routines like cleaning, lubrication and inspection voluntarily. As a result, they may understand more about their equipment and identify the common reasons for their machines to suffer from breakdowns.
- (iii) **Planned Maintenance (Keikaku-Hozen)** – Schedule preventive maintenances for every equipment in the company. It can be planned accordingly to the historical failure rate of each machine. This pillar can significantly reduce the frequency of equipment breakdown and unplanned downtime.
- (iv) **Quality Maintenance** – Minimize the number of defective products. Defects can be one of the major sources of production waste if the rejection rate is high. It is advised to design a system to detect defective products in the early stage of production so that it does not continue to waste more resources in the following

procedure. Root cause analysis method like fish bone diagram can help production engineers to identify and eliminate the sources of defect effectively.

- (v) **Early Equipment Management** – Apply the concept of maintenance prevention into the design of new equipment during planning, construction and installation phase. Maintenance prevention is a concept of minimizing maintenance fee and deterioration loss in the new equipment by introducing modifications in design based on the past maintenance data and experience. (Heymans, 2017). As a result, the frequency and difficulty of maintenance for the new equipment can be greatly reduced.
- (vi) **Training and Education** – Organize skill development program for all employees. Employee who is experienced and knowledgeable in regard to his work field is very essential to the growth and development of a company. Proper trainings should be given to help the employees make up for their knowledge gaps that is required to perform their daily duties and achieve TPM goals.
- (vii) **Safety, Health and Environment** – Create a safe and healthy working environment. Ananth (2017) suggested that in a safe working environment, workers are more likely to deliver a better performance and pose a good attitude towards their jobs.
- (viii) **TPM in Administration** – Apply TPM principles mentioned above in the office works. Implementation of TPM should not stop at manufacturing department, it should be practiced as a corporate culture in order to achieve “Zero” waste in an organization.

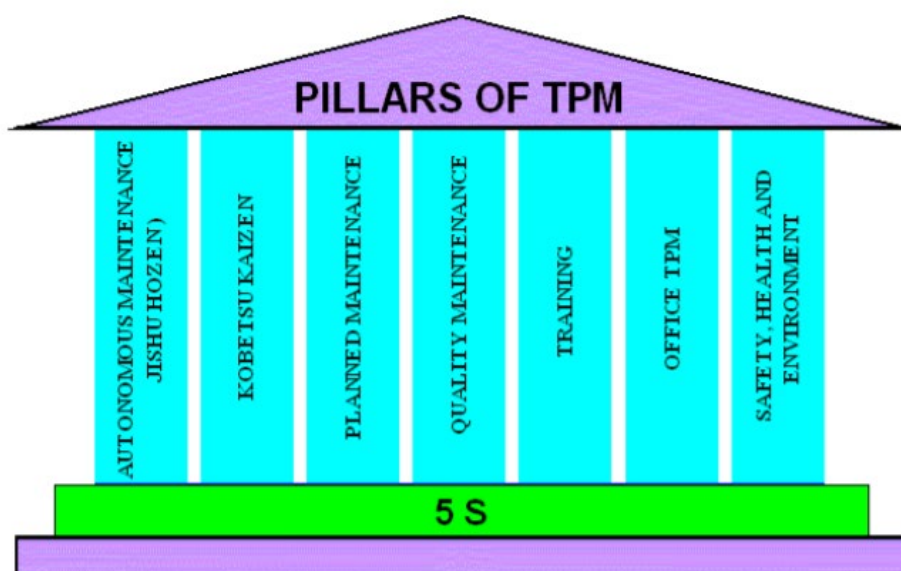


Figure 2.3: Eight Pillars of TPM (Venkatesh, 2005)

### 2.2.3 Correlation between OEE and TPM

The main objective of TPM is to enhance the productivity of a manufacturing firm by utilizing production equipment till its maximum capacity (Ljungberg, 1998). OEE is a performance metric introduced by Nakajima (1988) to measure equipment's efficiency (Chan, et al., 2005). Ahuja and Khamba (2008) claimed that there is a clear linkage between OEE and TPM, as OEE usually serves as an indicator to evaluate the final outcome of TPM implementation in a manufacturing firm.

### 2.3 OEE Implementation

There are many reasons for a production company to implement OEE. For instance, OEE can be used to identify production waste and measure the effect of improvement program. Other than that, it can act as a benchmark in the manufacturing industry.

Dr. Garg and Gupta (2012) presented the implementation of OEE in an automobile manufacturing company, XYZ in their journal. Before 1998, XYZ did not put any great effort in maintenance works. After that, due to intense competition in the market, XYZ started to adopt TPM in order to cut cost and improve productivity. Several Kaizen programs and preventive maintenance practices like lubrication and regular cleaning were carried out on the selected

machines. In order to measure the effect of TPM initiatives, Dr. Garg and Gupta (2012) used OEE to compare the difference in equipment performance between, before and after TPM implementation. As shown in Table 2.2, OEE scores for all the selected machines increase significantly after TPM implementation. Template used to compute OEE for Broaching Machine I is shown in Table 2.3.

Table 2.2: OEE Score Before and After TPM Implementation (Dr. Garg and Gupta, 2012)

Name of Machine	OEE Score (%)	
	Before TPM	After TPM
Broaching Machine I	59	70
Broaching Machine II	60	69
Cylindrical Grinder	53	67
Surface Grinder	50	65

Table 2.3: OEE Template for Broaching Machine I

After TPM Implementation		
A	Shift Time	450
B	Planned Downtime	60
C	Running Time (A-B)	390
D	Running Time Losses	58
E	Operating Time (C-D)	332
F	<b>Availability</b> (E/C) x 100%	85.1%
G	Output	207
H	Machine Speed (No of Pcs/min)	0.75
I	Expected Output	249
J	<b>Performance</b> (G/I) x 100%	83.1%
K	Rejection Quantity	2
L	<b>Quality</b> (G-K)/G x 100%	99%
	<b>OEE = F x J x L = 70%</b>	

Similar to the case above, Abhishek, et al. (2012) also used OEE to measure the improvement after TPM implementation in a pipe manufacturing company, M/S JAI MATA DI INDUSTRIES. Table 2.4 shows the OEE calculation template designed by Abhishek, et al. (2012).

Table 2.4: OEE Calculation Template (Abhishek, et al., 2012)

Machine/Factors	M1 (2 inch) 32-30 feet,1kg	M2 (3inch)24- 25feet, 1kg	M3 (4inch)18- 20feet,1kg	M4 (5inch)14- 15feet,1kg	M5 (6inch)13- 12feet,1kg
Production in (kg)	44467kg	55332kg	49108kg	49970kg	49908kg
Rejection in (kg)	1101kg	2000kg	1091kg	1110kg	1101kg
Final Goods in (kg)	43366kg	53332kg	48017kg	48860kg	48807kg
Break Down Time (hrs)	89hrs.	95.9%	85hrs.	89hrs.	82hrs.
Availability	64.7%	84.21%	87.2%	89.1%	88.04%
Performance Rate in %	81.2%	96.38%	83.35	82.9%	84.5%
Quality Rate in %	97.4%	77.87%	97.6%	97.8%	97.7%
OEE in %	72%	72.1%	71.7%	71.1%	72.8%

Herry, Farida and Lutfia (2018) used OEE to identify production loss in an automotive company, PT. TMI. They utilized the concept of OEE to analyze the six big losses in their daily production. Table 2.5 shows the analysis of six big losses in PT. TMI, Indonesia.

Table 2.5: Six Big Losses Analysis (Herry, Farida and Lutfia, 2018)

No	Six big losses	Total time loss (minutes)	(%)	Cumulative (%)
1	Reduced speed loss	56411	53%	53%
2	Setup/adjustment loss	36472	35%	88%
3	Yield/scrap loss	10398	10%	98%
4	Equipment failures loss	2389	2%	100%
5	Idling and minor stoppage Loss	0	0%	100%
6	Rework loss	0	0%	100%
	<b>Total</b>	<b>105670</b>	<b>100%</b>	

Kumar and Soni (2014) had conducted a survey on OEE implementation in the manufacturing companies in India. There are 750 manufacturing companies in India participated in this survey. Respondents were selected based on three parameters, which are type of production, number of equipment and

the level of TPM implementation. Questionnaire was designed to reveal the OEE structure and the perspective about OEE among India production companies. According to the survey results, 69.3% of companies measured OEE weekly and 19.7% of companies did it quarterly. 85.8% of companies thought that availability, performance and quality factors are equally important. Instead of using weighted average, 84.3% of companies used straight average method for their OEE calculations. Table 2.6 shows the OEE structure in respondents' companies whereas Table 2.7 shows the perspective of respondents about OEE.

Table 2.6: OEE Structures in India Production Companies (Kumar and Soni, 2014)

S. no.	Items	Options	Percentage of respondents	Cumulative percentage
1.	Frequency of OEE measure	Daily	2.8	2.8
		Weekly	69.3	72.1
		Monthly	4.7	76.8
		Quarterly	19.7	96.5
		Annually	3.5	100
2.	Ideal Cycle time (ICT)	Least CT in the group	16.5	16.5
		Designed CT	2.8	19.3
		CT defined by company	80.7	100
3.	At what level company calculate OEE	Product	2.8	2.8
		Process	2.8	5.6
		Equipment	79.5	85.1
		System	5.5	90.6
		Factory	9.4	100
4.	Which one factor is most important	Availability	6.7	6.7
		Performance	0.4	7.1
		Quality	7.1	14.2
		All equally important	85.8	100

Table 2.7: Perspectives of India Production Companies on OEE (Kumar and Soni, 2014)

S. no.	Items	Options	Percentage of respondents	Cumulative percentage
1.	Perspective of OEE	Productivity enhancement	1.2	1.2
		Equipment utilization	42.5	43.7
		To identify bottlenecks	1.2	44.9
		To purchase new facility	4.6	49.5
		All	50.5	100
2.	In what units company formulates OEE	Number of parts	2.8	2.8
		Volume	23.2	26
		Time	74	100
		Mix	0	100
3.	Method for OEE calculation	Straight average	84.3	84.3
		Weighted average	11.4	95.7
		Others	4.3	100

It is observed that most of the OEE implementations have a manual data collection system. Along with Industry Revolution 4.0, it is essential to develop

an automatic data collection system with IoT capability to obtain the real-time data. Giegling, et al. (1997) developed an automatic data acquisition system, PROMIS™ to track the work in progress in a semiconductor factory of Motorola. They used PROMIS™ to record the number of wafers produced and import the collected data to Capacity Utilization Bottleneck Efficiency System (CUBES) for further analysis. CUBES is an OEE framework developed by Konopka (1996) to study the manufacturing bottleneck of a production line. Figure 2.4 shows the flow of data from equipment into CUBES whereas Figure 2.5 shows the OEE template introduced by Konopka (1996).

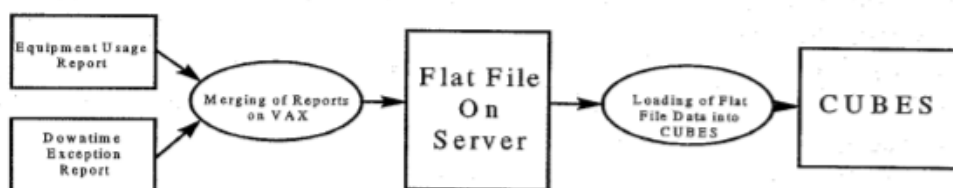


Figure 2.4: Data Flow at the Production Site of Motorola (Giegling, et al., 1997)

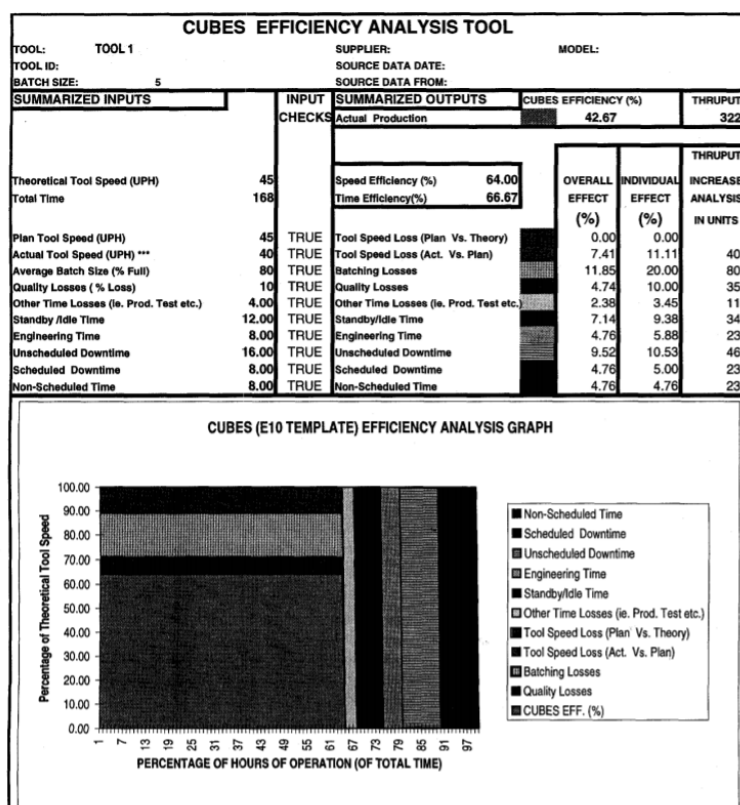


Figure 2.5: CUBES E10 Template (Konopka, 1996)



## 2.4 Plastic Injection Process

In order to propose an OEE measuring method for safety helmet plastic injection process, it is vital to understand its working principle. Plastic injection moulding is a manufacturing technology introduced by an American, John Wesley Hyatt in 1872. It is a process of injecting molten plastics into the mould cavity (Douglas, 1998). Figure 2.6 shows an injection moulding machine used to manufacture thermoplastic products. According to Andrew (2001), the process cycle of plastic injection moulding can be divided into four main stages, which are:

- (i) Clamping – Mould is closed firmly by the clamping unit to avoid material leakage during injection process. Sufficient clamping force is essential to ensure a good quality of end product. Clamping force is normally powered by a hydraulic motor.
- (ii) Injection – Firstly, plastic granules in hopper are fed into the rotating screw. After that, plastic granules are heated into molten plastic by the heaters surround at the rotating screw. Molten plastic is then pushed towards nozzle by the radial force and axial force produced by the rotating screw. When its injection pressure hits the preset parameter, molten plastic will be injected into the mould cavity.
- (iii) Cooling – Cooling occurs when molten plastic makes contact with the mould surfaces. There are many cooling channels inside the mould. Typical cooling mediums used in injection moulding process are water and oil.
- (iv) Ejection – After molten plastic solidified into the desired shape, mould will be opened for ejection. Final product will be ejected from the mould core through ejector pins.

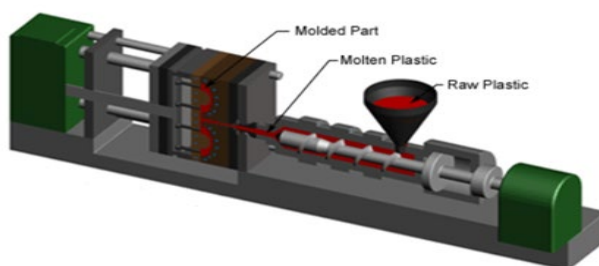


Figure 2.6: Injection Moulding Machine

## CHAPTER 3

### METHODOLOGY AND WORK PLAN

#### 3.1 Project Flow

From initiation to closure, this project was planned to be completed within two long semesters. It can be divided into two parts, as illustrated in Figure 3.1.

First and foremost, a background study on OEE was carried out to provide a basic understanding before problem formulation. Keywords like OEE, TPM, IoT and DAQ in the FYP title were identified and a further study was conducted to master all the relevant concepts. Few visits were paid to Proguard Technologies (M) Sdn Bhd for a better understanding of research problem and industry operation. Objectives of this project were defined by using SMART approach. Journals and articles that are related to OEE and TPM were reviewed and their discussions were summarised into Chapter 2. A Microsoft Excel template was designed to perform OEE analysis at Proguard Technologies (M) Sdn Bhd. Graphical representation will be generated automatically in the OEE template to assist management level in the identification of major production losses. Towards the end of FYP Part I, a conceptual design of automatic data acquisition system with IoT capability was proposed to implement on the safety helmet plastic injection process. All progress in FYP Part I were documented into progress report. Figure 3.2 shows the project's Gantt Chart of FYP Part I.

In FYP Part II, a schematic electrical circuit diagram was drawn to show the wire connection between sensors, actuators and microcontroller. Subsequently, a program was developed to allow the communications between microcontroller and other modules. After code debugging, prototype was fabricated by using 3D printing. Trial run of prototype was carried out at home due to Covid pandemic. Modifications and improvement were made to increase the reliability of the proposed solution. All design considerations were properly documented in this thesis report for future reference. Figure 3.3 shows the project's Gantt Chart of FYP Part II.

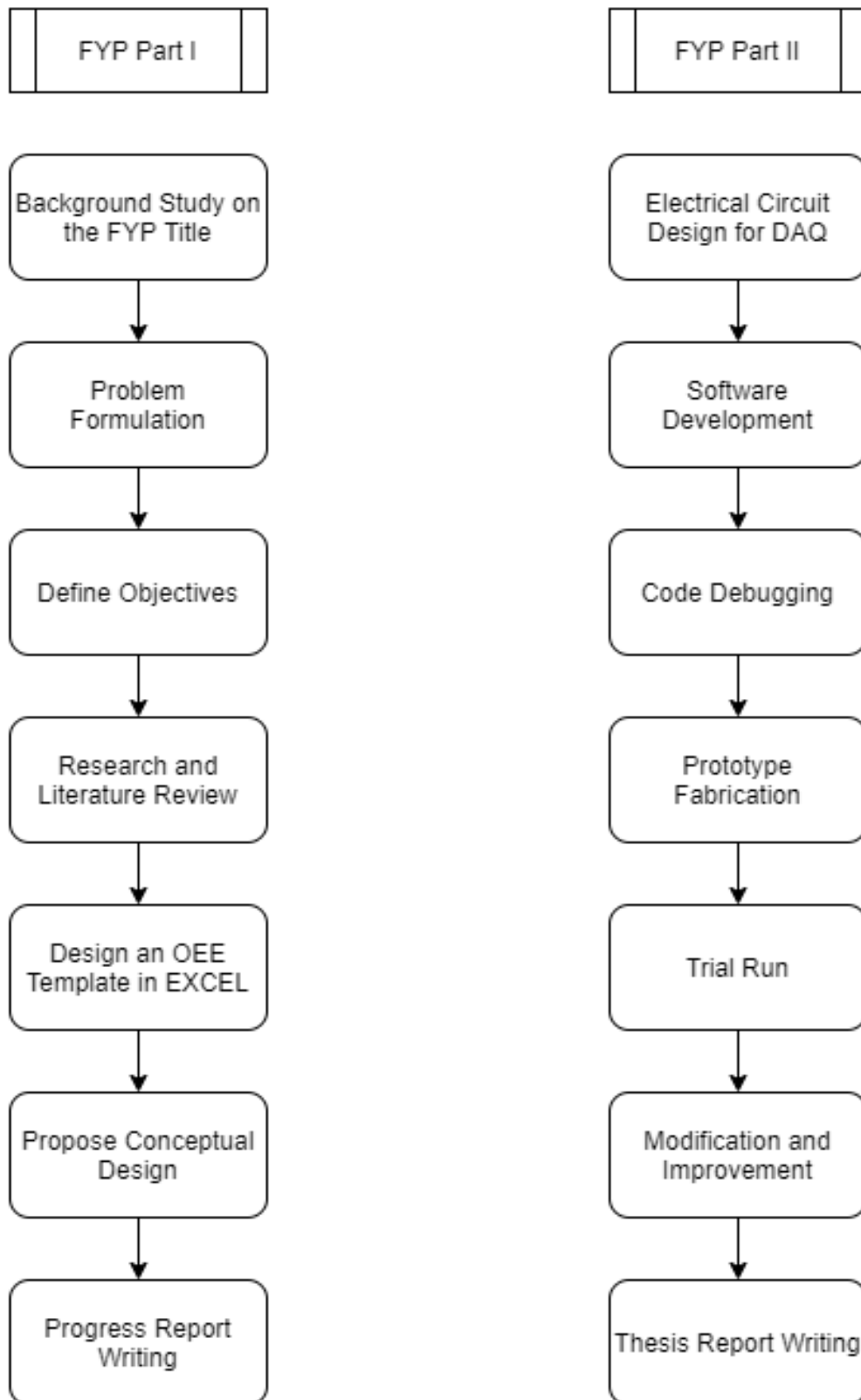


Figure 3.1: Project Flow Chart

**Gantt Chart (Part-1 / ~~Part-2~~) (please strike out whichever is not applicable)**

No.	Project Activities	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
M1	Problem Formulation & Project Planning														
M2	Literature Review														
M3	Conceptual Design														
M4	Report Writing & Presentation														

Figure 3.2: Gantt Chart of FYP Part I

**Gantt Chart (Part-1 / Part-2) (please strike out whichever is not applicable)**

No.	Project Activities	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
M1	Software Development	■	■	■	■										
M2	Prototype Fabrication				■	■	■								
M3	Modification and Improvement						■	■	■	■	■				
M4	Thesis Report Writing								■	■	■	■	■	■	■
M5	Presentation														■

Figure 3.3: Gantt Chart of FYP Part II

### 3.2 Customize OEE Template

Figure 3.4 depicts the customize OEE computation template created in Microsoft Excel. It was used to compute the OEE score of plastic injection machines at Proguard Technologies (M) Sdn Bhd. Cells in pink colour represent the scores for three OEE factors while cell in yellow colour shows the final OEE score.

LINE 1									
Operator Name:	1)				Filled By				
	2)				Checked By				
	3)				Date		12/4/2021		
<b>Availability Score :</b>					<b>0</b>		<b>%</b>		
Total No. of Shift				2	PER DAY				
Shift Duration				8	H	0	MIN		
OT Duration				4	H	0	MIN		
Total Planned Production Time				24	H	0	MIN		
Unplanned Downtime					Planned Downtime				
Equipment Failure	0	H	0	MIN	Machine Setup	0	H	0	MIN
Others		H		MIN	Mould Changeover	0	H	0	MIN
Total	0	H	0	MIN	Machine Maintenance	0	H	0	MIN
					Others	0	H	0	MIN
					Total	0	H	0	MIN
<b>Performance Score :</b>					<b>0</b>		<b>%</b>		
Cycle Time				1	MIN / PART				
Total No. of Units Produced				0	PCS				
<b>Quality Score :</b>					<b>0</b>		<b>%</b>		
No. of Rejected Products				0	PCS				
<b>O E E Score :</b>					<b>0</b>		<b>%</b>		

Figure 3.4: Customize OEE Computation Template in Ms Excel

### 3.3 Conceptual Design

Figure 3.5 depicts the block diagram of the proposed data acquisition system. The proposed design uses ESP32 as microcontroller to communicate with input buttons, sensors, actuators, output display and IoT server.

As shown in Figure 3.5, input side is mainly comprised of sensors and push buttons. They are responsible for collecting the required data to perform OEE calculations. Ultrasonic sensor is used to count the total number of units produced whereas load cell sensor is used to measure the weight of a safety helmet. There are two push buttons used in this design. One is allocated to count the total number of rejected products while another one is used to initiate the downtime counter in ESP32 microcontroller.

On the other hand, output consists of liquid crystal display, LEDs, buzzer as well as stepper motor. Liquid crystal display is used to show important information such as date, total number of units produced and total number of rejected products. Three LEDs represent three different parameters. Green, red and blue LEDs are used to indicate wifi connection, error and inspection status respectively. Besides, buzzer is adopted to alert operator when rejection mechanism is running. Stepper motor is chosen to power up the rejection mechanism.

The proposed data acquisition system is designed to equip with IoT capability. Thus, ESP32 will send the collected data to IoT server, ThingSpeak for further processing and analysis. In ThingSpeak, all logged data will be presented in graphical approach. Email with OEE report will be sent to production engineers daily. A call alert will be generated automatically when the OEE score falls below than 40%. Engineers and management can monitor the OEE score of every production line via ThingShow in their smart phones at anytime and anywhere. In addition, production dashboard is intended to replace white board in displaying important information like production schedule, maintenance schedule, output target and etc.

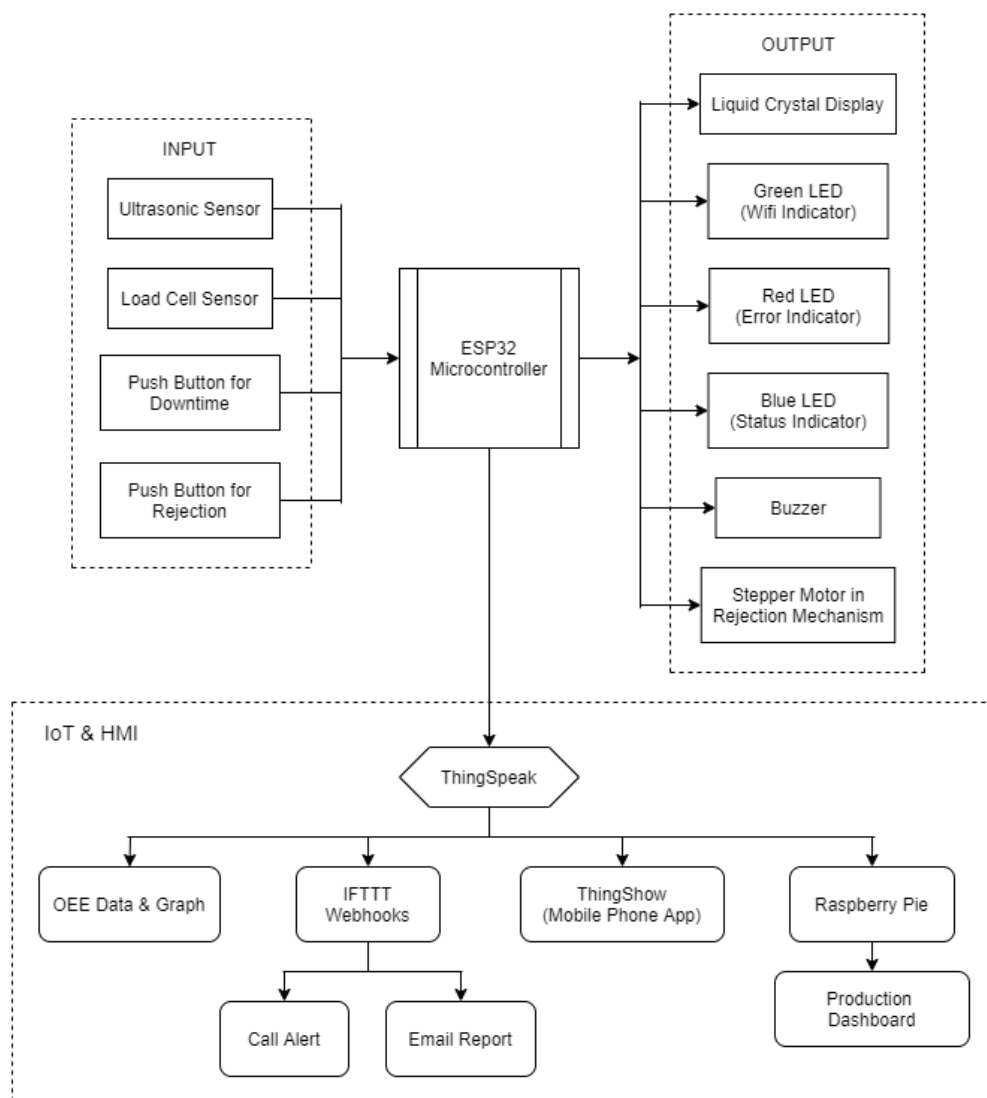


Figure 3.5: Block Diagram of DAQ System

### 3.4 Mechanical Design

All mechanical design and 3D modelling tasks in this project were completed by using a CAD software, named SolidWorks. In this section, all design considerations on mechanical structure will be discussed. Figure 3.6 presents the isometric view of the proposed data acquisition system whereas Figure 3.7 shows its top view with section labelling.





Figure 3.6: Isometric View of The Proposed DAQ System

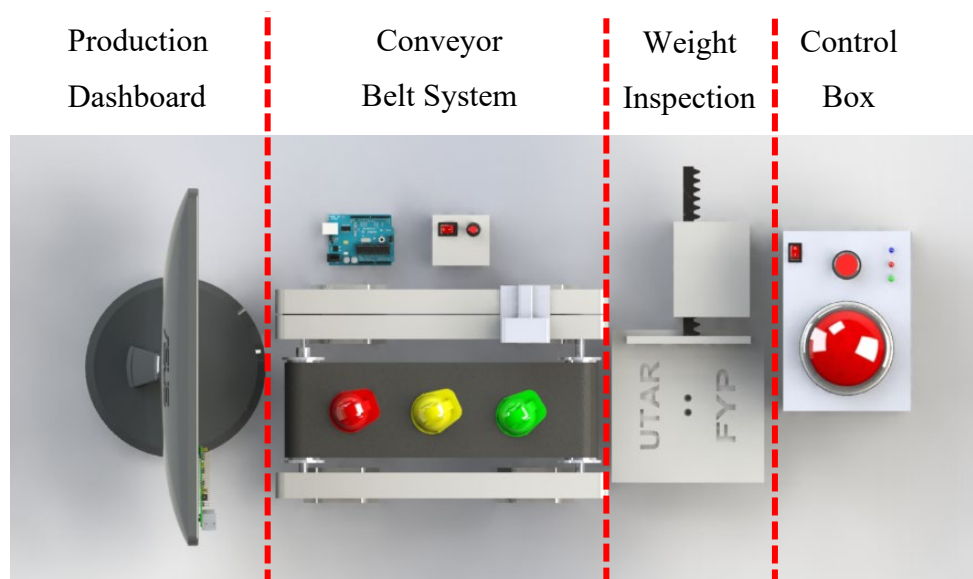


Figure 3.7: Top View of The Proposed DAQ System

### 3.4.1 Conveyor Belt System

Conveyor belt system is necessary to ensure a smooth transition of products between every stages. In this project, a conveyor belt system was designed to transport safety helmet from plastic injection machine to inspection area. Figure 3.8 depicts the exploded view of conveyor belt system while Table 3.1 shows its Bill of Materials (BOM).

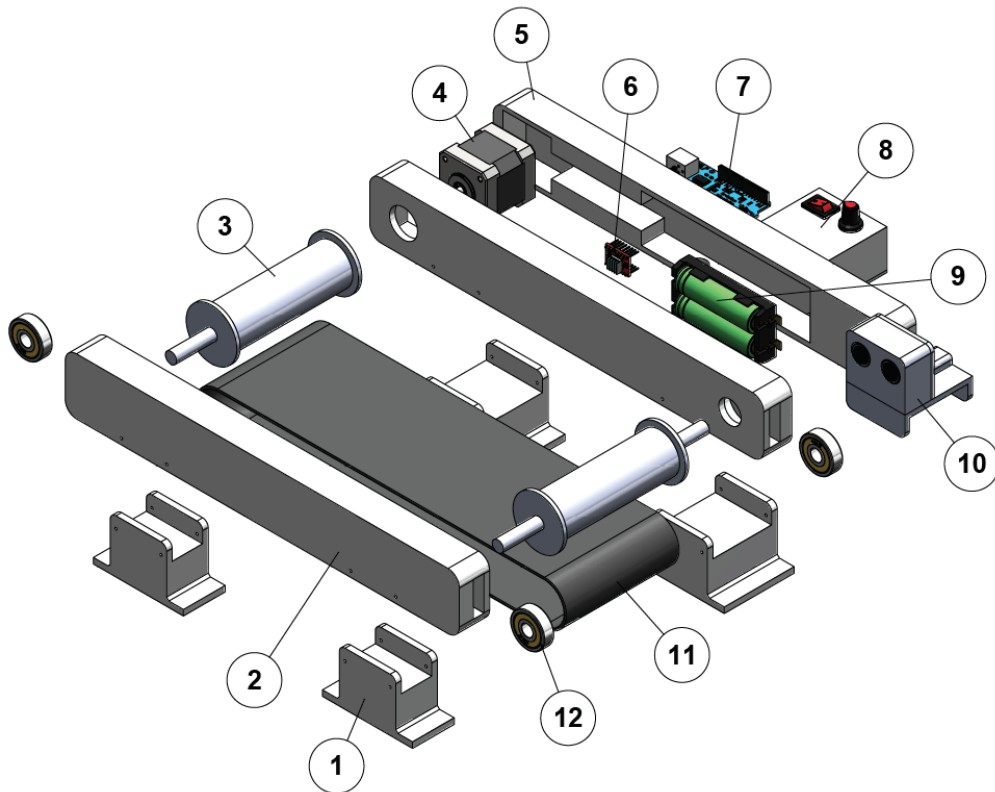


Figure 3.8: Exploded View of Conveyor Belt System

Table 3.1: BOM of Conveyor Belt System

Item No	Part Name	Quantity
1	Supporting Base	4
2	Left Frame Structure	1
3	Conveyor Pulley	2
4	NEMA 17 Stepper Motor	1
5	Right Frame Structure	1
6	A4988 Stepper Motor Driver	1
7	Arduino UNO	1
8	Conveyor Speed Controller	1
9	18650 Lithium Ion Battery	2
10	HC-SR04 Ultrasonic Sensor	1
11	Conveyor Belt	1
12	6200ZZ Ball Bearing	3

Concurrently, an ultrasonic sensor was mounted at the side of conveyor belt system to count the total number of safety helmets produced.

### 3.4.2 Weight Inspection and Rejection Mechanism

Weight inspection is an essential process to ensure all safety helmet have a consistent quality. In this project, bar type load cell is used to measure the weight of a safety helmet. Supporting base and measuring plate are needed to create space for bar type load cell to perform deformation when it is under certain load. Rejection mechanism is simply powered by a stepper motor. Pinion and rack are used to convert the rotational motion of stepper motor shaft into a linear motion to reject faulty products. Figure 3.9 shows the exploded view of weight inspection module and rejection mechanism whereas Table 3.2 shows the BOM for them.

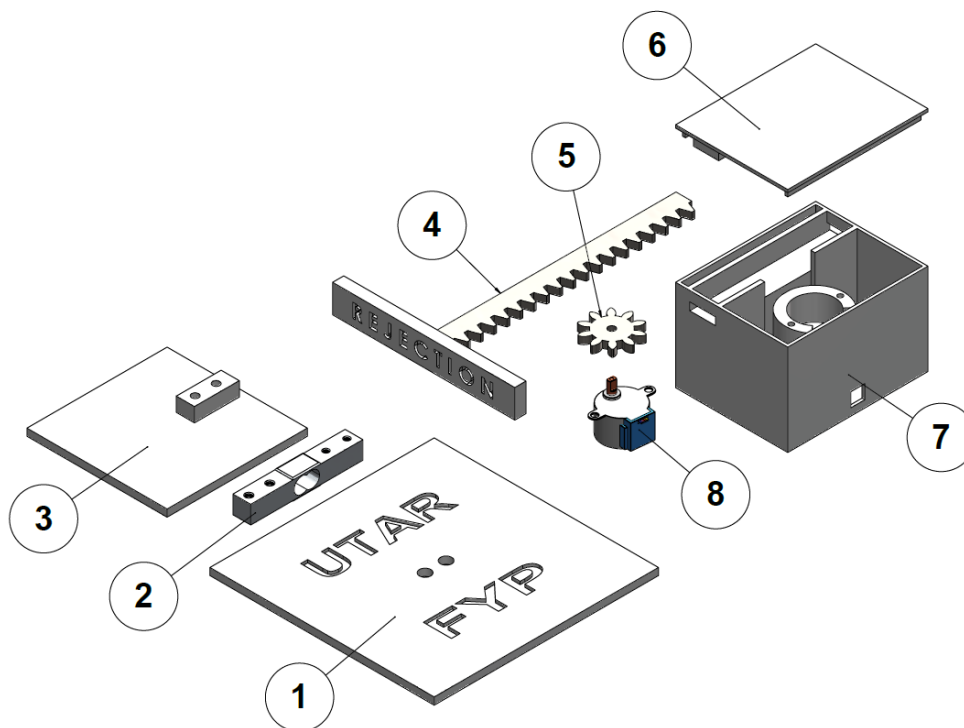


Figure 3.9: Exploded View of Weight Inspection Module & Rejection Mechanism

Table 3.2: BOM of Weight Inspection Module &amp; Rejection Mechanism

Item No	Part Name	Quantity
1	Measuring Plate	1
2	5kg Load Cell Bar	1
3	Supporting Base	1
4	Linear Rack with Pushing Plate	1
5	Spurs Gear	1
6	Top Cover for Gear Box	1
7	Gear Box	1
8	28BYJ-48 Stepper Motor	1

### 3.4.3 Control Box

Control box is a sort of physical interface which allows user to communicate directly with its machine or equipment. In this project, a control box is built to allow operator to control and monitor the performance of data acquisition system. Like any other control box, this control box is comprised of microcontroller, LED indicator, LCD display, emergency button as well as on-off switch. It has an overall dimension of 190mm x 135mm x 140mm. Figure 3.10 shows the exploded view of control box while Table 3.3 shows its BOM.

Table 3.3: BOM of Control Box

Item No	Part Name	Quantity
1	Housing of Control Box	1
2	HX711 Amplifier Module	1
3	ULN2003 Stepper Motor Driver Board	1
4	100mm Big Dome Push Button	1
5	14500 Lithium Ion Battery	2
6	Top Cover of Control Box	1
7	2-Pin Red Rocker Switch	1
8	5mm Green LED	1
9	5mm Red LED	1
10	5mm Blue LED	1

11	22mm Arcade Push Button	1
12	ESP32	1
13	Piezo Active Buzzer	1
14	1602 Liquid Crystal Display with I2C Module	1

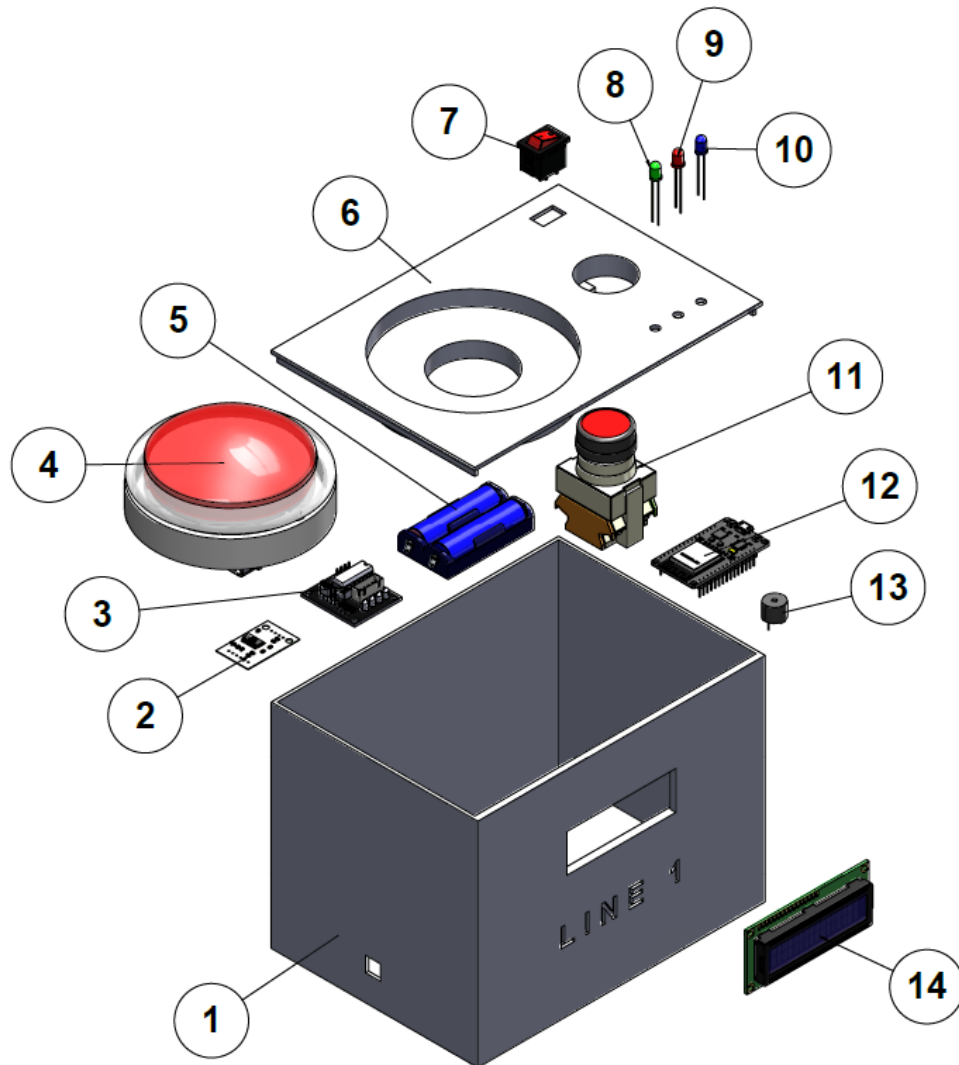


Figure 3.10: Exploded View of Control Box

### 3.5 Electrical and Electronics

Electrical circuit design was completed by using Fritzing and Proteus softwares. Fritzing is used to draw graphical wiring diagram shown in Figure 3.11 whereas Proteus is used to construct schematic circuit diagram shown in Figure 3.12. In this section, every electronics component and their roles in data acquisition system will be discussed.

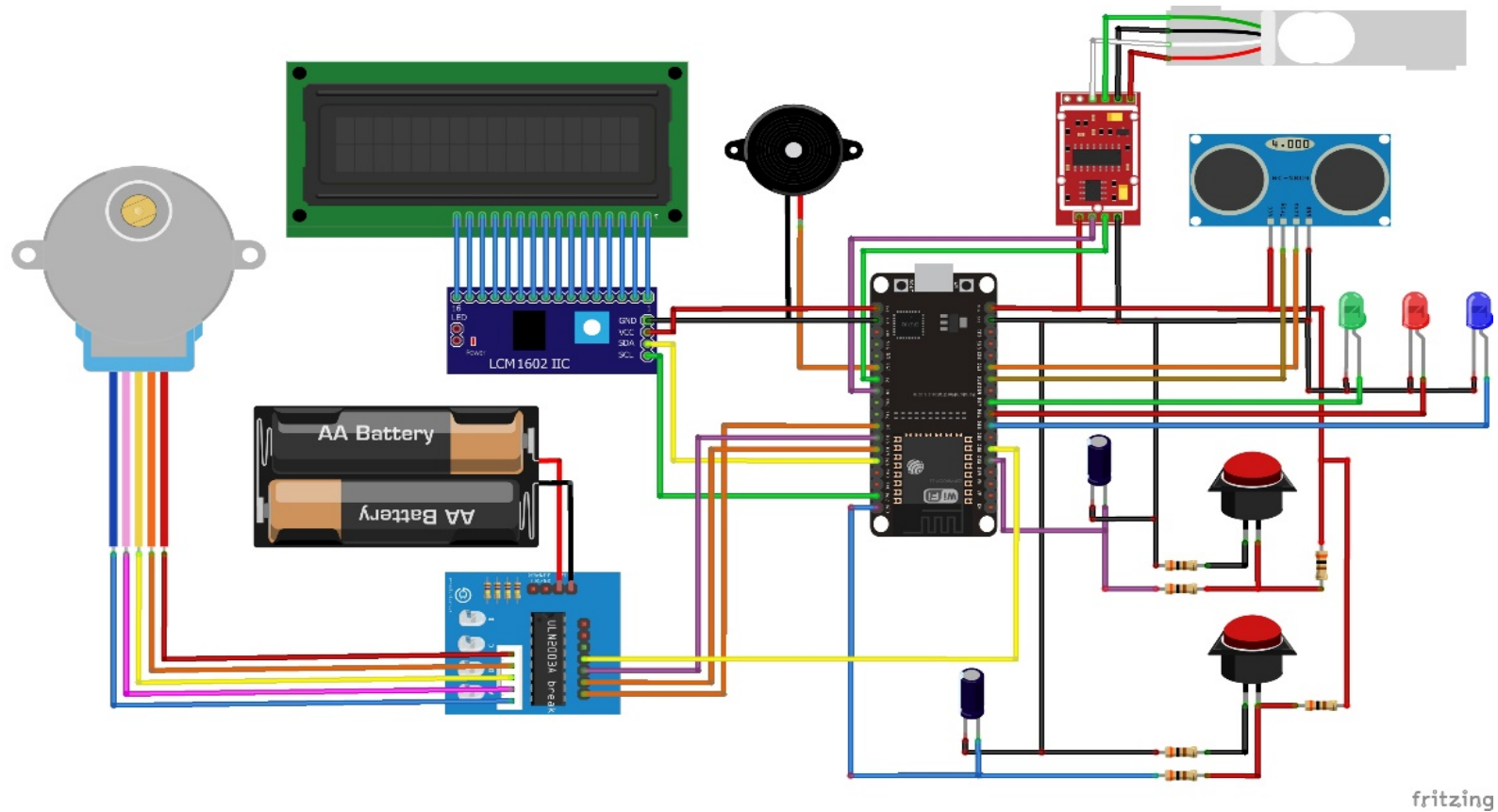


Figure 3.11: Graphical Wiring Diagram of DAQ System



In the proposed design of data acquisition system, ESP32 has two important roles, which are sending collected data to IoT server as well as communicating with connected sensors and actuators.

Table 3.4: Specifications of ESP32 Wroom-32 (Espressif Systems, 2014)

Number of Core	2 (Dual Core)
Wi-Fi	2.4 GHz (150 Mbits/s)
Bluetooth	Bluetooth Low Energy and Legacy Bluetooth
Architecture	32 Bits
Maximum Clock Frequency	240 MHz
RAM	512 kB
Operating Voltage	2.2V to 3.6V
Number of Pin	30
Number of ADC Channel	18 (12 Bit Resolution)
Number of DAC Channel	2
Number of PWM Output	16

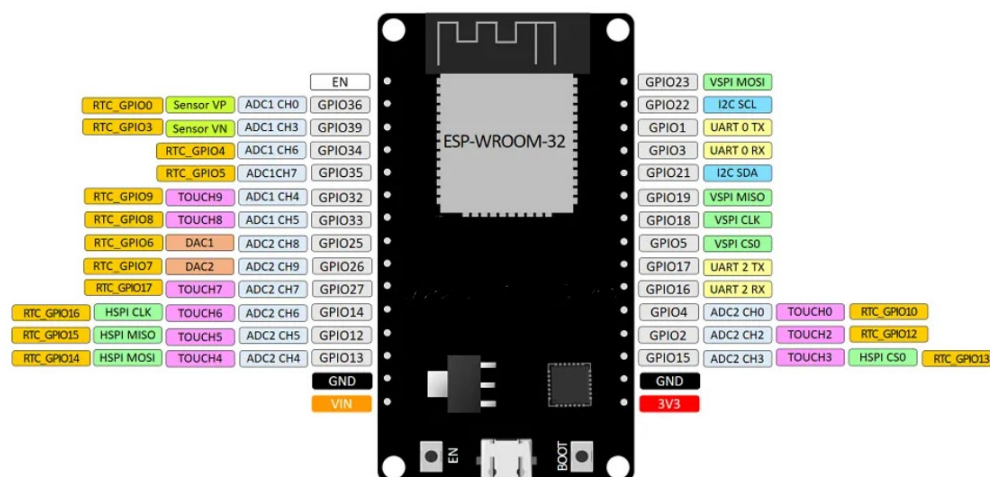


Figure 3.13: ESP32 Wroom-32 Pin Layout (Rui and Sara, 2020)

### 3.5.2 HC-SR04 Ultrasonic Sensor

HC-SR04 Ultrasonic sensor is a popular sensor for object detection. It consists of one transmitter that emits ultrasonic sound waves and one receiver that picks



up reflected waves if there is an object blocks its emitting waves. It has four pins, which are named Vcc, Trig, Echo and Gnd. Trig pin is connected to the output pin of ESP32 to trigger ultrasonic sound pulses whereas Echo pin is connected to the input pin of ESP32 to receive an electrical signal when ultrasonic waves are reflected. Vcc and Gnd pins are used to power up this sensor. Figure 3.14 shows the pinout of HC-SR04 ultrasonic sensor while Table 3.5 shows its specifications. HC-SR04 ultrasonic sensor is chosen for this project because it is economical (RM2.90), simple to use and low-powered.

In the data acquisition system, ultrasonic sensor plays a role of counting the total number of safety helmet produced. It is one of the input data that is necessary to perform OEE calculations.

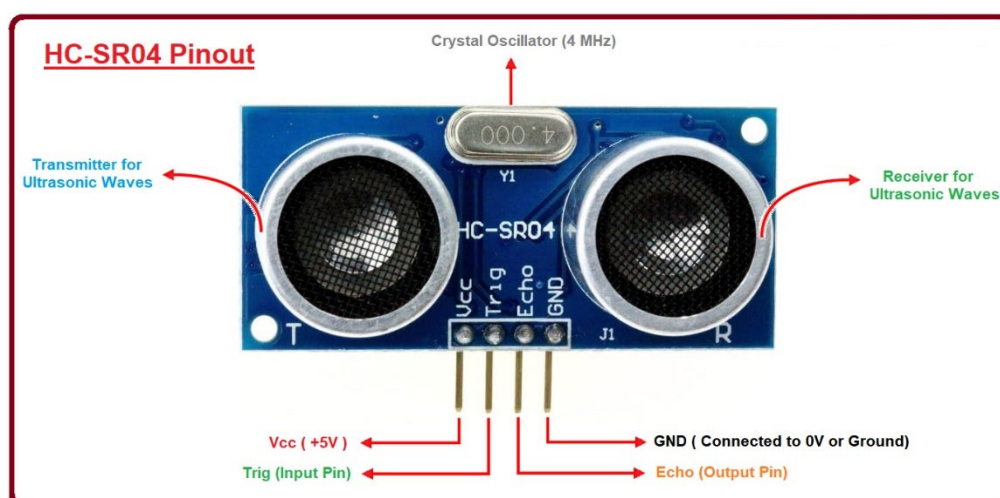


Figure 3.14: Pinout of HC-SR04 Ultrasonic Sensor (Adnan, 2018)

Table 3.5: Specifications of HC-SR04 Ultrasonic Sensor (Rui and Sara, 2013)

Operating Voltage	5V DC
Operating Current	15mA
Operating Frequency	40kHz
Ranging Distance	2cm to 400cm
Resolution	3mm
Trigger Input Pulse Width	10 $\mu$ S

### 3.5.3 5kg Load Cell with HX711 Amplifier

Load cell is a transducer that can convert weight quantity into an electrical output signal. Bar type strain gauge load cell is used in this project. In order to measure the bending distortion at aluminium bar, strain gauges are installed in wheatstone bridge configuration to detect a very small voltage change across output terminal. HX711 amplifier board is required to amplify the output voltage signal so that it is measurable to ESP32. Figure 3.15 shows the 5kg load cell with HX711 amplifier whereas Table 3.6 depicts the load cell specifications.

Bar type strain gauge load cell (Sparkfun TAL220B) is chosen for this project due to its high sensitivity (1mV/V), high accuracy (combined error of  $\pm 0.05\%$  FS) and low cost (RM19). In the data acquisition system, load cell plays a role of measuring the weight of safety helmet and returning the measured values back to ESP32.

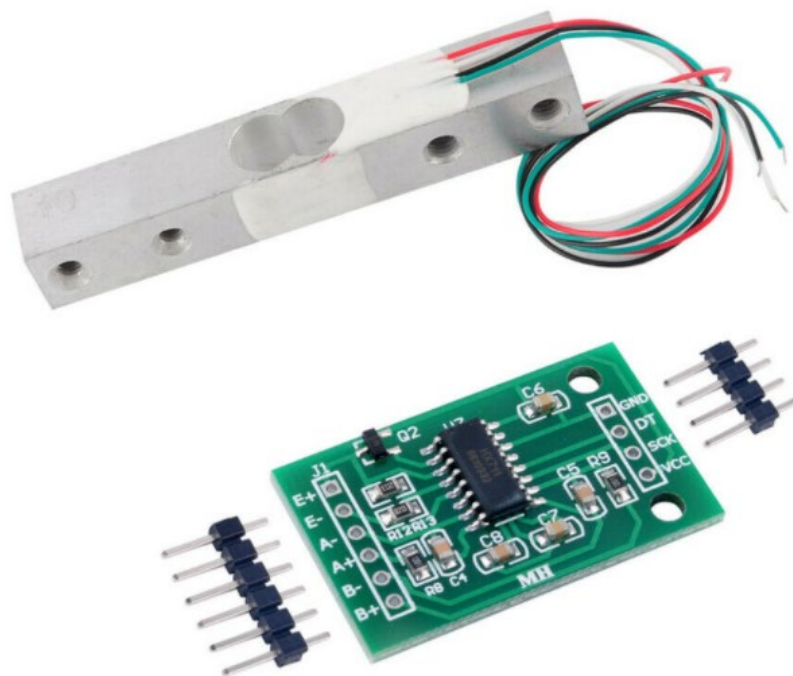


Figure 3.15: 5kg Load Cell with HX711 Amplifier (Sarah, 2016)

Table 3.6: Specification of Sparkfun TAL220B Load Cell

Capacity	5kg
Safe Overload	120% FS
Ultimate Overload	150% FS

Excitation Voltage	3V to 10V DC
Non-linearity	$\pm 0.05\%$ FS
Repeatability	$\pm 0.05\%$ FS
Operating Temperature Range	-10°C to +55°C

### 3.5.4 Momentary Push Button

Momentary push button is a switch that will cause temporary change in an electrical circuit. Spring in the push button will make the switch returns to its original state after applied force is removed. Two momentary push buttons are used in this project to allow user communicates with ESP32 when there is an unscheduled downtime or a rejected product due to plastic defects.

### 3.5.5 Active Buzzer

Active buzzer is an audio signalling device which can turn audio signals into sound signals when it is connected to a DC power supply. Sound is generated by the vibration of ceramic disc in buzzer. It consists of two pins, one is connected to power while another one is connected to ground. Figure 3.16 shows the picture of an active buzzer. In the proposed data acquisition system, buzzer acts as an output to provide alert signal when there is an emergency case.



Figure 3.16: Active Buzzer

### 3.5.6 16 x 2 Liquid Crystal Display with I2C Module

Liquid crystal display (LCD) is a kind of flat-panel display that uses polarized light to form images. It is composed of several layers such as glass filter, liquid crystal layer, electrodes and polarized films. Light is projected from background and amount of light that projects onto cover glass can be controlled by applying electric current to the layer of liquid crystal. 16 x 2 stands for 16 columns and 2 rows. There are 16 pins for connection in 16 x 2 LCD model. However, 16-pin communication can be reduced to 4-pin communication by using i2c module. Figure 3.17 shows the LCD model used in this project.

LCD is chosen for this project because it is cheap (RM10.50), easy access and highly programmable. In the data acquisition system, it is placed on the control box to display some important information like date and time, total number of units produced, rejected amount and etc.

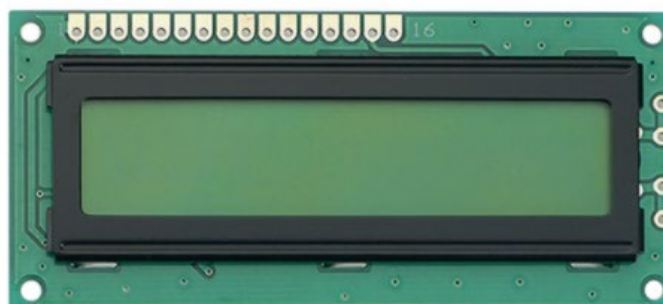


Figure 3.17: 16 x 2 Liquid Crystal Display

### 3.5.7 28BYJ-48 Stepper Motor with ULN2003 Driver Board

28BYJ-48 Stepper motor is a DC electric motor that revolves in discrete steps. It comes with four coils and the coils are required to be energized in a logical sequence in order to perform a rotation. This control can be easily achieved through programming with the help of ULN2003 driver board. Figure 3.18 shows the image of 28BYJ-48 Stepper motor and ULN2003 Motor driver while Table 3.7 lists out the motor specifications.

It is selected for this project because it has the advantages of low cost (RM11.80), precise positioning, precise speed control and excellent torque at low speeds. In the proposed system, it is used to power the rejection mechanism to push away safety helmets that do not meet weight requirement.



Figure 3.18: 28BYJ-48 Stepper Motor with ULN2003 Driver

Table 3.7: 28BYJ-48 Stepper Motor Specifications

Rated Voltage	5V DC
Number of Phases	4
Stride Angle	5.625° or 64 steps per revolution
DC Resistance	50Ω ± 7%
Pull In Torque	300 gf.cm
Noise Level	<35dB

### 3.6 Software Development

Software development in this project can be divided into three groups, which are main program in control box, OEE computation program in IoT server as well as low-code program to run the live dashboard. Three of them are interconnected to each other but working as separate programs. This is because three of them are developed by different programming environments.

Main program in ESP32 Microcontroller is developed by using Arduino IDE while OEE computation program in IoT server is developed via a numeric computing platform called MATLAB. Lastly, low-code program to run the live dashboard is developed through the built-in programming tool in Raspberry Pi, called Node-RED. Figure 3.19 shows the framework of the proposed data acquisition system.

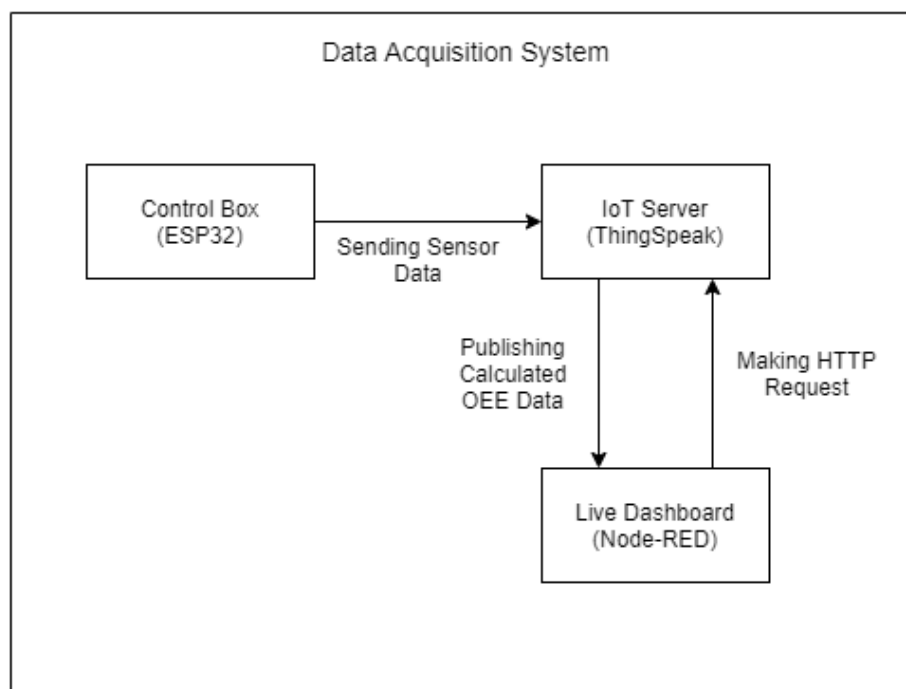


Figure 3.19: DAQ System Framework

### 3.6.1 Main Program in ESP32 Microcontroller

Features to be achieved by main program are:

- (i) Communicate with various types of sensor and collect input data for OEE calculation.
- (ii) Establish a wireless connection and publish collected data to IoT server, ThingSpeak.
- (iii) Reject safety helmets that do not meet weight requirement.

Figure 3.20 shows the flowchart of main program. First and foremost, main program will attempt to connect to a wireless network. After wifi connection is established, it will enable external interrupt and obtain the information of date and time from NTP server. Before entering loop cycle, it will check load cell connection and calibrate load cell with a preset value. Next, main program will enter a loop cycle of reading sensor output and sending data to IoT server.

Besides this, Figure 3.21 and Figure 3.22 depict the flowcharts of external interrupt for rejection and downtime push buttons respectively. When external interrupt is triggered, main program will stop the execution of loop cycle and enter interrupt service routine to perform the respective operations.

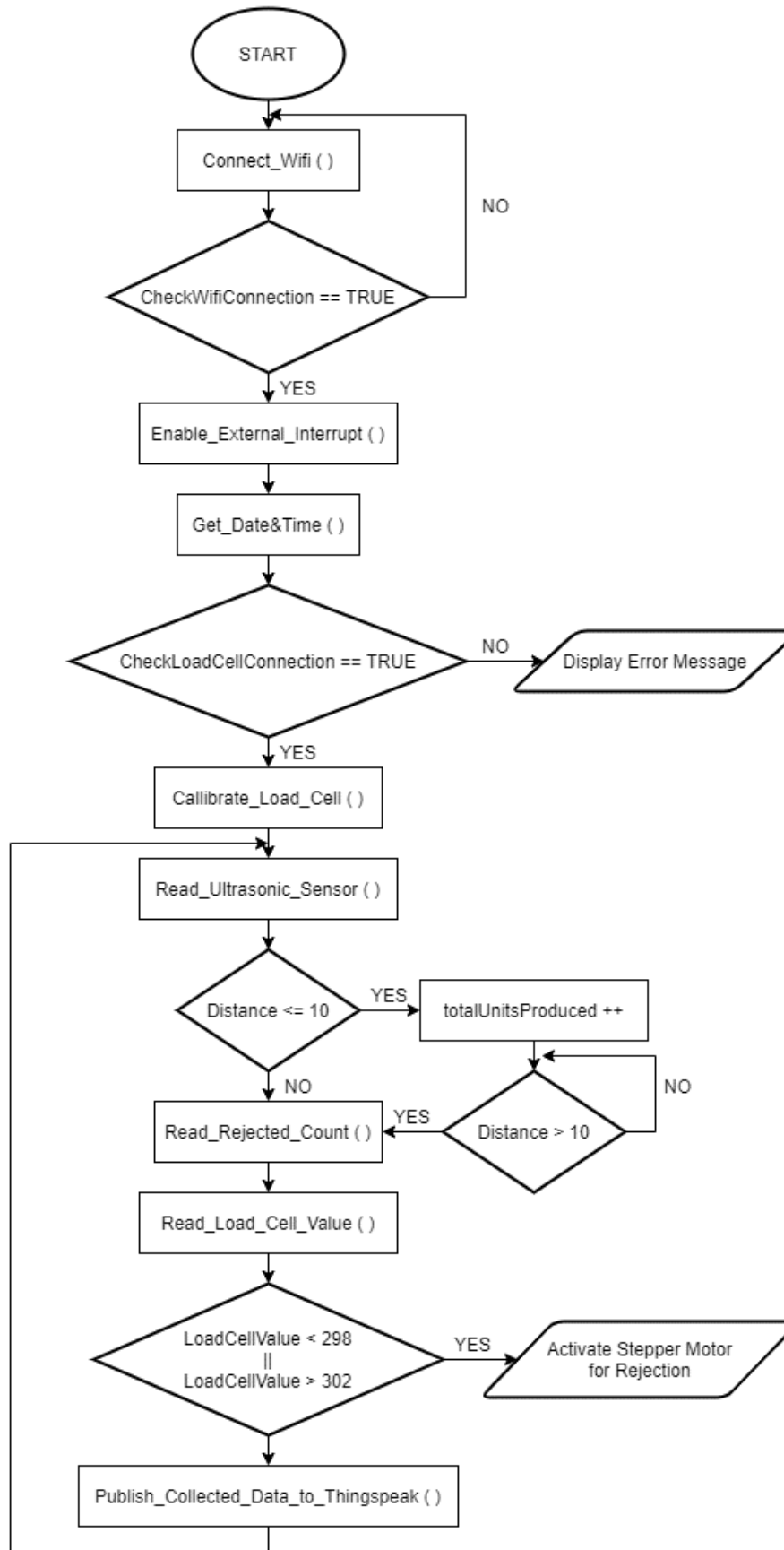


Figure 3.20: Main Program Flowchart

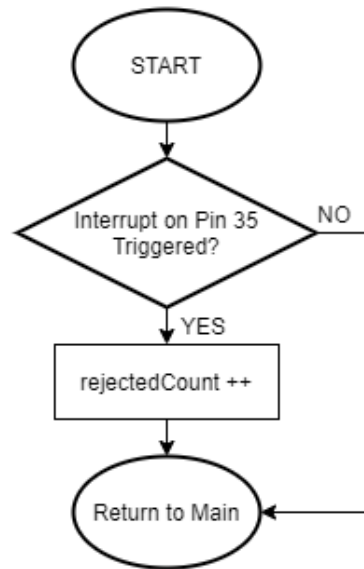


Figure 3.21: Flowchart of External Interrupt for Rejection Push Button

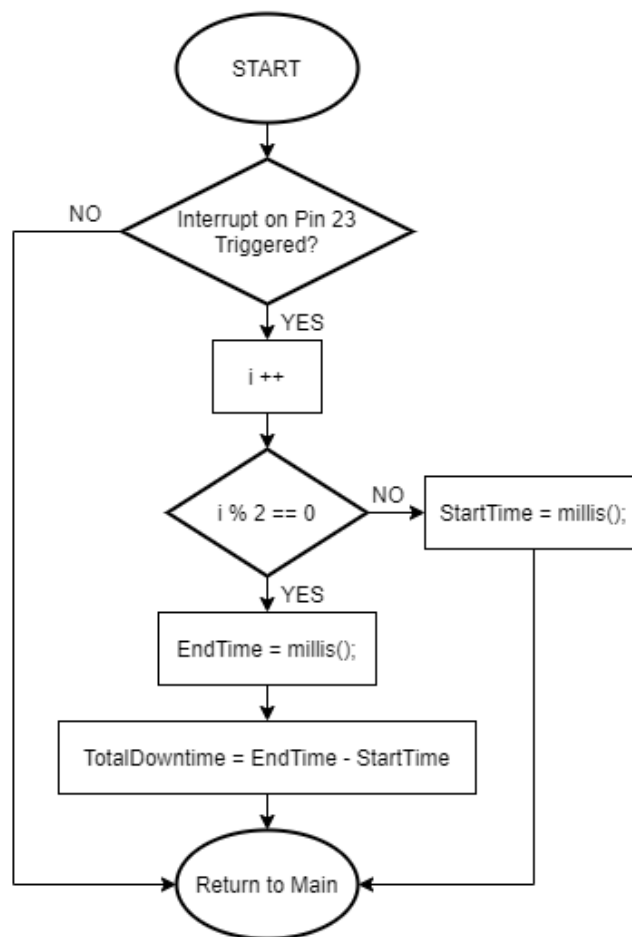


Figure 3.22: Flowchart of External Interrupt for Downtime Push Button



### 3.6.1.1 Establishment of Wireless Connection

Figure 3.23 shows the code to establish a connection to wireless network. Library of <WiFi.h> is included so that main program can attempt to connect to Wifi with a simple function called WiFi.begin. This function requires two input parameters which are network username and its respective password. A while loop is created to prevent main program from proceeding to next section without a successful connection to wireless network. In addition, it is coded so that a green LED will light up when it is connected to Wifi.

```
#include <WiFi.h> //Wifi Library

char ssid[] = "kexinchua95@unifi"; //Network Username
char pass[] = "01311100"; //Password

WiFi.begin(ssid, pass); //Establishing connection to wireless network
```

Figure 3.23: Code Segment for Connecting to Wireless Network

### 3.6.1.2 Counting Total Number of Safety Helmet Produced

Figure 3.24 shows the code segment for counting total number of safety helmet produced. An ultrasonic sensor is used to detect the presence of safety helmet on conveyor belt. Firstly, a pulse of 10 $\mu$ s is applied to the Trig pin of ultrasonic sensor. As a result, it will send out a sonic burst at 40kHz. PulseIn function is then called to check if Echo pin receives any reflected ultrasonic wave. If there is no reflected signal received after 38ms, Echo pin will return low to indicate no object is placed within the sensing area.

A variable named totalUnitsProduced is declared to act as a counter. When a safety helmet is detected by ultrasonic sensor, the variable will update its stored value by adding one to the current value. A do-while loop is created to avoid counter from adding more than one time for only one safety helmet.

```

do
{
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    duration = pulseIn(echoPin, HIGH);
    distance = duration*0.034/2;
}while (distance <= 10);

```

Figure 3.24: A Do-while Loop for Counting Total Number of Safety Helmet Produced

### 3.6.1.3 External Interrupts for Rejection and Downtime Counter

Instead of polling, external interrupt is used to minimize the waiting time of loop cycle. Millis() function is used to record the interrupt time and a conditional if logic is created to prevent main program from performing more than one time of interrupt in a short time. Figure 3.25 shows the millis() function used in external interrupts. A function called attachInterrupt() is required to use to configure an external interrupt during setup. Figure 3.26 depicts the way of enabling an external interrupt.

```

interrupt_time = millis();

if (interrupt_time - last_interrupt_time > 5000)
{
    rejectedcount = rejectedcount + 1;
    last_interrupt_time = interrupt_time;
}

```

Figure 3.25: Millis() Function used in External Interrupts

```

void setup()
{
    pinMode(rej_button, INPUT);
    attachInterrupt(35, isr, FALLING); //Enable External Interrupt

    pinMode(downtime_button, INPUT);
    attachInterrupt(23, isr2, FALLING); //Enable External Interrupt
}

```

Figure 3.26: Code Segment to Enable External Interrupts

### 3.6.1.4 Measuring Weight of Every Safety Helmet

Figure 3.27 shows the code segment for measuring load cell value. Library of <HX711\_ADC.h> is included to ease the communication between ESP32 and HX711 amplifier. LoadCell.setCalFactor() can be used to calibrate load cell with a preset value during setup. LoadCell.update() is called to check for new data while LoadCell.getData() is used to obtain the current value measured by load cell. It is found that HX711 amplifier is very sensitive to noises, thus a code is written as shown in Figure 3.28 to ensure a proper wire connection on load cell during setup. A conditional if statement is created to reject any safety helmet that does not meet the preset weight requirement. Figure 3.29 shows the code to control stepper motor for rejection. In order to avoid injury and unnecessary contact, buzzer will beep to alert operator during rejection process. Figure 3.30 shows the code to activate buzzer.

```
#include <HX711_ADC.h> //Load Cell Library

LoadCell.setCalFactor(calibrationValue); //Load Cell Calibration

LoadCell.update();
i = LoadCell.getData();
```

Figure 3.27: Code Segment for Measuring Weight

```
if (LoadCell.getTareTimeoutFlag()) {
  Serial.println("Timeout, check MCU>HX711 wiring and pin designations");
  while (1);
}
```

Figure 3.28: Code Segment for Checking Load Cell Connection

```
if (weight < 298 || weight > 302){ //Weight Requirement
  rejectedcount = rejectedcount + 1;
  stepper.setSpeed(15);
  stepper.step(1250); //Rejection
  stepper.step(-1250);
}
```

Figure 3.29: Code Segment for Rejection

```

for (int x = 0; x <5; x++){
    ledcWriteTone(0,1000); //Buzzer beeps to alert operator
    delay(500);
    ledcWriteTone(0,0);
    delay(500);
}

```

Figure 3.30: Code to Activate Buzzer

### 3.6.1.5 Publishing Collected OEE Data to ThingSpeak

Figure 3.31 shows the code to send the collected OEE data to IoT platform, ThingSpeak. Library of “ThingSpeak.h” is included so that data can be published to ThingSpeak server with a function called ThingSpeak.writeFields(). A conditional if statement is created to avoid the incident of sending same data twice to ThingSpeak server.

```

#include "ThingSpeak.h" //ThingSpeak Library

void loop()
{
    if (totalUnitsProduced != last_totalUnitsProduced || rejectedcount
        != last_rejectedcount || weight != last_weight || TotalDowntime !=
        LastTotalDowntime)
    {
        ThingSpeak.setField(1,totalUnitsProduced);
        ThingSpeak.setField(2,rejectedcount);
        ThingSpeak.setField(3,weight);
        ThingSpeak.setField(4,TotalDowntime);
        ThingSpeak.writeFields(myChannelNumber,myWriteAPIKey);
    }
}

```

Figure 3.31: Code Segment for Publishing Collected OEE Data to ThingSpeak

### 3.6.2 OEE Computation Program in IoT Server

An OEE data post-processing program is created to analyse the data retrieved from ESP32. By implementing this program, we are able to know the OEE score of each production line. Figure 3.32 shows the flowchart of OEE computation program running in ThingSpeak server. Formula to calculate OEE score, availability index, performance index as well as quality index are shown in Equation 2.1, 2.2, 2.3 and 2.4 respectively. Calculation results will be published back to ThingSpeak channel before program ends. OEE analysis is designed to be carried out on a daily basis thus a timer interrupt is created to reset all

parameters back to zero at 12 a.m. everyday. Figure 3.33 depicts the timer interrupt created in ThingSpeak server.

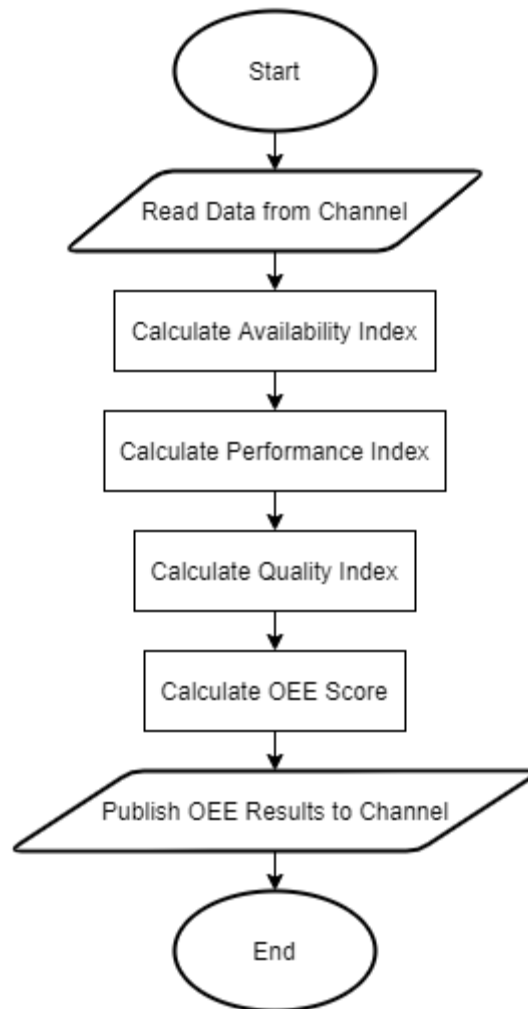


Figure 3.32: Flowchart of OEE Analysis

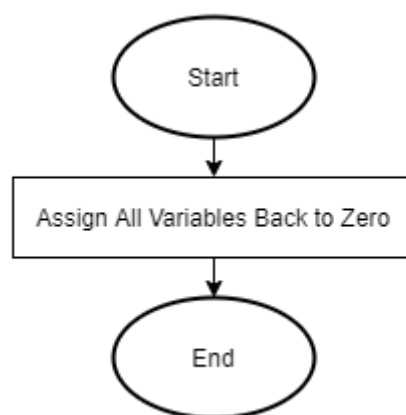


Figure 3.33: Flowchart of Timer Interrupt

### 3.6.3 Low-code Program in Production Dashboard

A low-code program is constructed to run a live dashboard at production site. Figure 3.34 shows the low-code program created in Node-RED. Firstly, raspberry pi will send a http request to ThingSpeak server to acquire the information of production. Messages returned from ThingSpeak server are all in the form of JavaScript Object Notation (JSON). In order to access the message properties, they are required to be converted back to JavaScript object. This conversion can be done by placing a JSON node in the low-code program. After conversion, values in the returned messages are ready to be extracted for display. Several widgets such as donut gauge and numeric display box are used for a better illustration purpose.

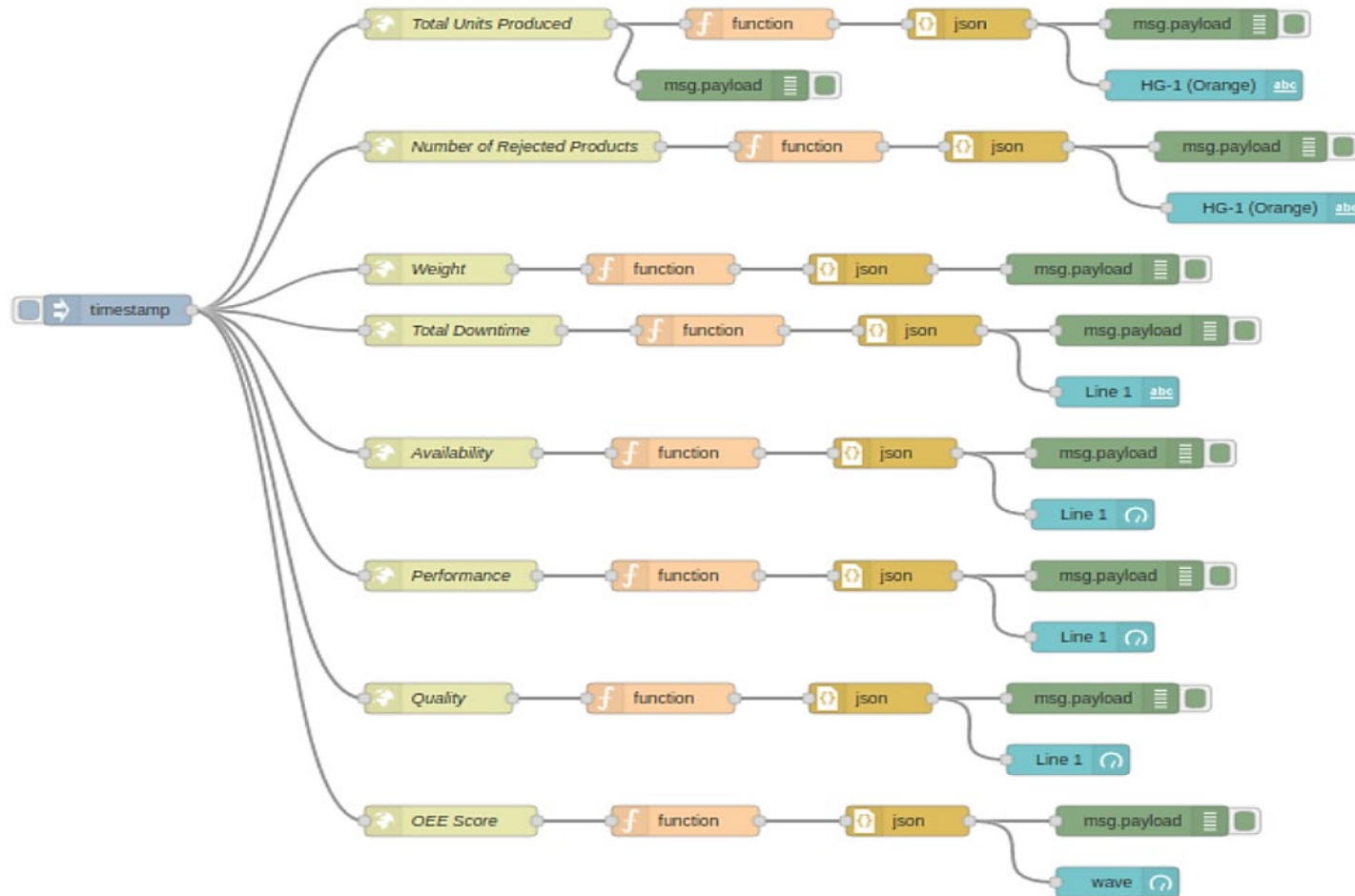


Figure 3.34: Low-code Program in Node-RED

## CHAPTER 4

### RESULTS AND DISCUSSIONS

#### 4.1 Manual Data Collection System

A visit was paid to Proguard Technologies (M) Sdn Bhd on 2<sup>nd</sup> of April 2021 to collect data required for OEE calculations. Input data required to compute OEE score are total planned production time, actual production time, total number of units produced, product cycle time as well as total number of rejected products. Total planned production time is normally fixed by planner at management level. In Proguard Technologies (M) Sdn Bhd, there are two normal shifts of eight hours each in one day, along with two four-hour overtime shifts. Hence, equipment in Proguard Technologies (M) Sdn Bhd are running 24 hours a day, 7 days a week. Actual production time is the remaining production time after the subtraction of unplanned and planned downtime. Examples of downtime event are machine failure, machine setup, maintenance, changeover of mould and etc. An input sheet is created to ease the input process of data. Figure 4.1 shows the data collected from Line 1, 2 and 3 at Proguard Technologies (M) Sdn Bhd. Figure 4.2, Figure 4.3 and Figure 4.4 show the OEE calculations for Line 1, Line 2 and Line 3 respectively. Taking Line 1 as example, steps to compute OEE score of a production line are shown as follows.

$$\textit{Total Planned Production Time} = 24 \times 60 = 1440 \textit{ mins}$$

$$\begin{aligned} \textit{Actual Production Time} &= 1440 - (1 \times 60) - (2 \times 60) - (1 \times 60 + 30) \\ &= 1170 \textit{ mins} \end{aligned}$$

$$\begin{aligned} \textit{Availability} &= \frac{\textit{Actual Production Time}}{\textit{Planned Production Time}} \times 100\% \\ &= \frac{1170}{1440} \times 100\% = 81.25\% \end{aligned}$$



$$\begin{aligned}
 \text{Performance} &= \frac{\text{Total Number of Units Produced} \times \text{Cycle Time}}{\text{Actual Production Time}} \times 100\% \\
 &= \frac{1200 \times 0.8}{1170} \times 100\% = 82.05\%
 \end{aligned}$$

$$\begin{aligned}
 \text{Quality} &= \frac{\text{Number of Good Units}}{\text{Total Number of Units Produced}} \times 100\% \\
 &= \frac{1200 - 50}{1200} \times 100\% = 95.83\%
 \end{aligned}$$

$$\begin{aligned}
 \text{OEE Score} &= \text{Availability} \times \text{Performance} \times \text{Quality} \\
 &= \frac{81.25 \times 82.05 \times 95.83}{10000} = 63.89\%
 \end{aligned}$$

In the customize template, OEE score will be computed automatically once the required input data are inserted. Meanwhile, an OEE analysis chart will be generated for visualisation purpose. Figure 4.5, Figure 4.6 and Figure 4.7 depict the OEE analysis chart for Line 1, Line 2 and Line 3 respectively. According to OEE score ranking proposed by Singh (2018), performance of Proguard Technologies (M) Sdn Bhd is considered as typical manufacturers. Based on the OEE analysis charts, it is observed that performance factor is the most unstable factor among three OEE indexes. However, Proguard Technologies (M) Sdn Bhd showed a good performance in quality index. They managed to score more than 90% consistently in quality index for all three production lines. Figure 4.8 shows the OEE score indicators for Line 1, 2 and 3.

INPUT SHEET											
SHIFT INFORMATION											
Total No of Shift per Day		2									
Shift Duration		8	H	0	MIN						
OT Duration		4	H	0	MIN						
LINE INFORMATION											
LINE 1	Machine Failure Time		H		MIN	Machine Output					
	Machine Setup Time	1	H		MIN	Model No	ADV 1	Total No of Units Produced	200	Rejected Amount	13
	Machine Maintenance Time	2	H		MIN	Model No	ADV 2	Total No of Units Produced	300	Rejected Amount	15
	Mould Changeover Time	1	H	30	MIN	Model No	ADV 4	Total No of Units Produced	700	Rejected Amount	22
	Others		H		MIN	Model No		Total No of Units Produced		Rejected Amount	
							Model No		Total No of Units Produced		Rejected Amount
						<b>GRAND TOTAL</b>		1200			50
LINE 2	Machine Failure Time		H		MIN	Machine Output					
	Machine Setup Time	1	H		MIN	Model No	ALPIN PLUS	Total No of Units Produced	500	Rejected Amount	20
	Machine Maintenance Time	1	H	45	MIN	Model No	630V	Total No of Units Produced	250	Rejected Amount	8
	Mould Changeover Time	1	H		MIN	Model No	ADV 11	Total No of Units Produced	750	Rejected Amount	32
	Others		H		MIN	Model No		Total No of Units Produced		Rejected Amount	
							Model No		Total No of Units Produced		Rejected Amount
						<b>GRAND TOTAL</b>		1500			60
LINE 3	Machine Failure Time		H		MIN	Machine Output					
	Machine Setup Time	1	H		MIN	Model No	HHV 9	Total No of Units Produced	250	Rejected Amount	11
	Machine Maintenance Time	2	H		MIN	Model No	ADV 18	Total No of Units Produced	450	Rejected Amount	17
	Mould Changeover Time	1	H		MIN	Model No	ADV 3E	Total No of Units Produced	500	Rejected Amount	26
	Others		H		MIN	Model No		Total No of Units Produced		Rejected Amount	
							Model No		Total No of Units Produced		Rejected Amount
						<b>GRAND TOTAL</b>		1200			54

Figure 4.1: Data Collected on 2<sup>nd</sup> of April 2021 at Proguard Technologies (M) Sdn Bhd

LINE 1									
Operator Name:	1)				Filled By				
	2)				Checked By				
	3)				Date		13/4/2021		
<b>Availability Score :</b>					<b>81.25</b>		<b>%</b>		
Total No. of Shift				2	PER DAY				
Shift Duration				8	H	0	MIN		
OT Duration				4	H	0	MIN		
Total Planned Production Time				24	H	0	MIN		
Unplanned Downtime					Planned Downtime				
Equipment Failure	0	H	0	MIN	Machine Setup	1	H	0	MIN
Others		H		MIN	Mould Changeover	1	H	30	MIN
Total	0	H	0	MIN	Machine Maintenance	2	H	0	MIN
					Others	0	H	0	MIN
					Total	4	H	30	MIN
<b>Performance Score :</b>					<b>82.05</b>		<b>%</b>		
Cycle Time				0.8	MIN / PART				
Total No. of Units Produced				1200	PCS				
<b>Quality Score :</b>					<b>95.83</b>		<b>%</b>		
No. of Rejected Products				50	PCS				
<b>O E E Score :</b>					<b>63.89</b>		<b>%</b>		

Figure 4.2: OEE Calculations for Line 1

LINE 2									
Operator Name:	1)				Filled By				
	2)				Checked By				
	3)				Date		13/4/2021		
<b>Availability Score :</b>					<b>84.38</b>		<b>%</b>		
Total No. of Shift				2	PER DAY				
Shift Duration				8	H	0	MIN		
OT Duration				4	H	0	MIN		
Total Planned Production Time				24	H	0	MIN		
Unplanned Downtime					Planned Downtime				
Equipment Failure	0	H	0	MIN	Machine Setup	1	H	0	MIN
Others		H		MIN	Mould Changeover	1	H	0	MIN
Total	0	H	0	MIN	Machine Maintenance	1	H	45	MIN
					Others	0	H	0	MIN
					Total	3	H	45	MIN
<b>Performance Score :</b>					<b>98.77</b>		<b>%</b>		
Cycle Time				0.8	MIN / PART				
Total No. of Units Produced				1500	PCS				
<b>Quality Score :</b>					<b>96</b>		<b>%</b>		
No. of Rejected Products				60	PCS				
<b>O E E Score :</b>					<b>80.01</b>		<b>%</b>		

Figure 4.3: OEE Calculations for Line 2

LINE 3									
Operator Name:	1)				Filled By				
	2)				Checked By				
	3)				Date		13/4/2021		
<b>Availability Score :</b>					<b>83.33</b>		<b>%</b>		
Total No. of Shift				2	PER DAY				
Shift Duration				8	H	0	MIN		
OT Duration				4	H	0	MIN		
Total Planned Production Time				24	H	0	MIN		
Unplanned Downtime					Planned Downtime				
Equipment Failure	0	H	0	MIN	Machine Setup	1	H	0	MIN
Others		H		MIN	Mould Changeover	1	H	0	MIN
Total	0	H	0	MIN	Machine Maintenance	2	H	0	MIN
					Others	0	H	0	MIN
					Total	4	H	0	MIN
<b>Performance Score :</b>					<b>80</b>		<b>%</b>		
Cycle Time				0.8	MIN / PART				
Total No. of Units Produced				1200	PCS				
<b>Quality Score :</b>					<b>95.5</b>		<b>%</b>		
No. of Rejected Products				54	PCS				
<b>O E E Score :</b>					<b>63.66</b>		<b>%</b>		

Figure 4.4: OEE Calculations for Line 3

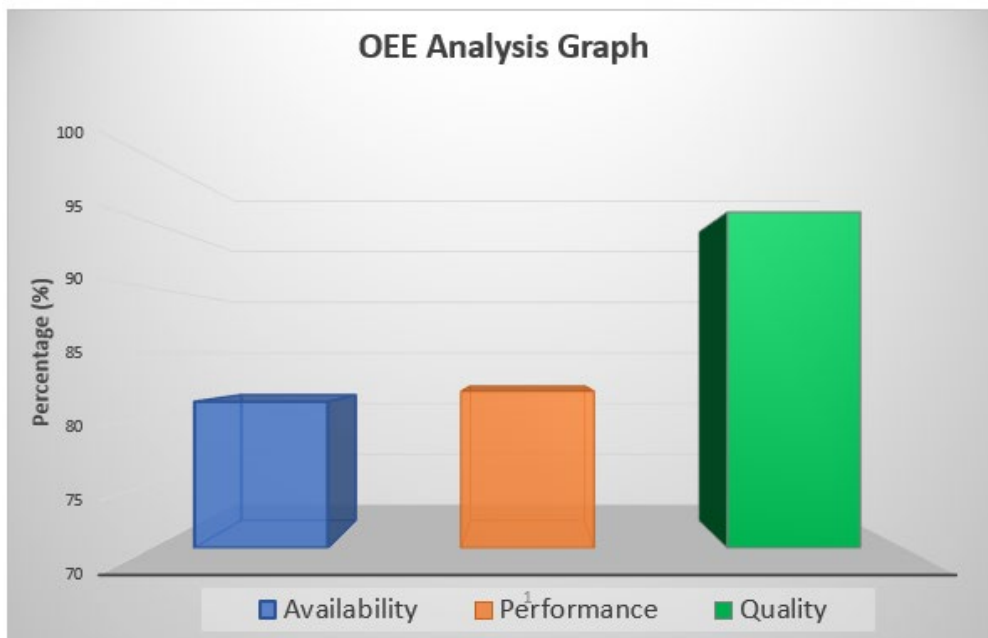


Figure 4.5: OEE Analysis Chart for Line 1

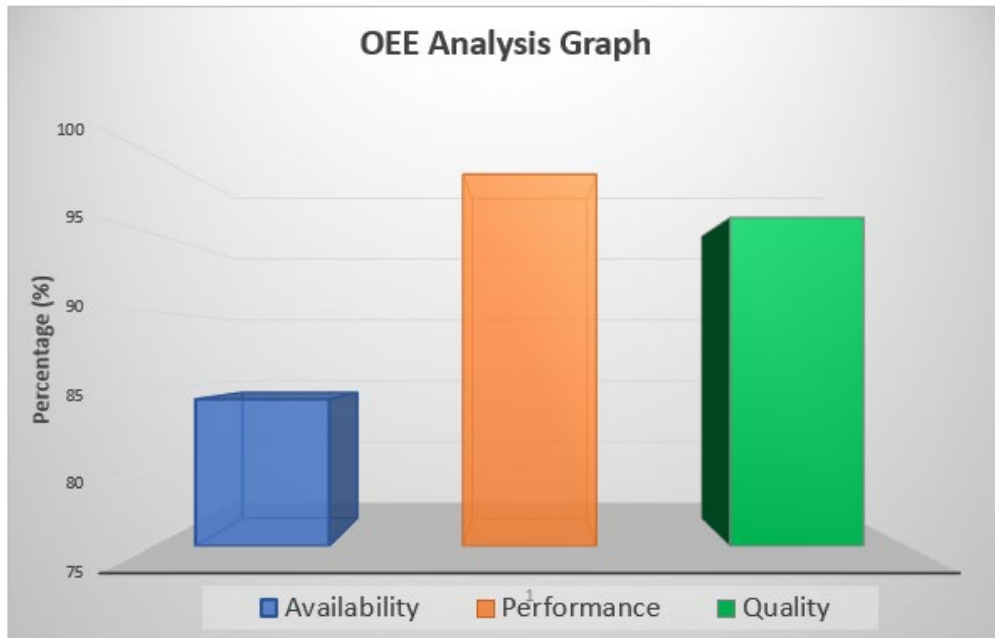


Figure 4.6: OEE Analysis Chart for Line 2

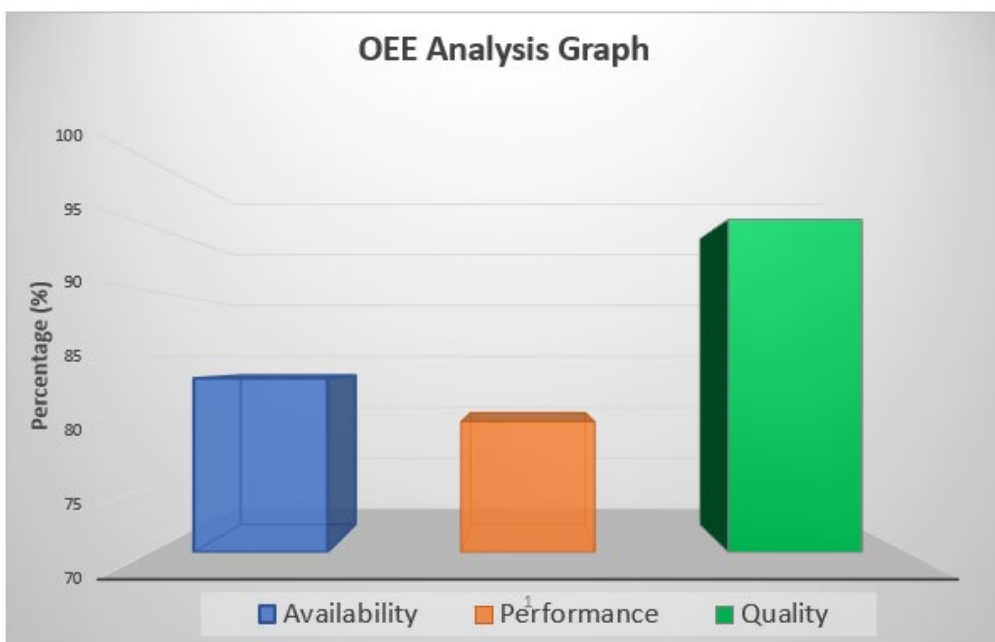


Figure 4.7: OEE Analysis Chart for Line 3

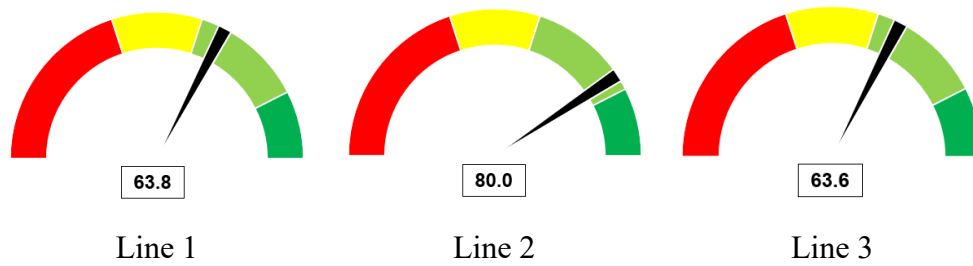


Figure 4.8: OEE Score Indicators

## 4.2 Fabrication of Prototype

A prototype of automatic data acquisition system with IoT capability was fabricated for demonstration purpose. In this section, mechanical fabrication and electrical circuit connection will be discussed. Figure 4.9, Figure 4.10 as well as Figure 4.11 show the side view, top view and front view of prototype respectively.

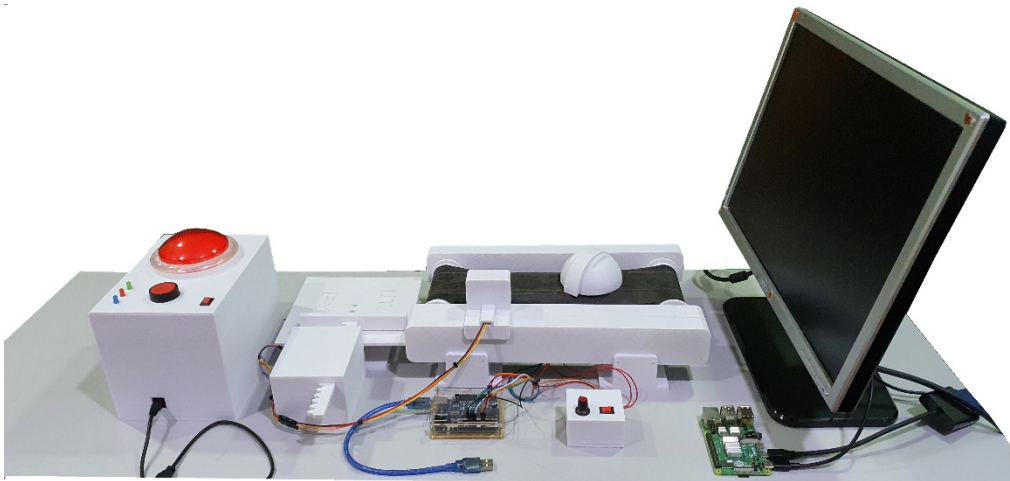


Figure 4.9: Side View of Prototype

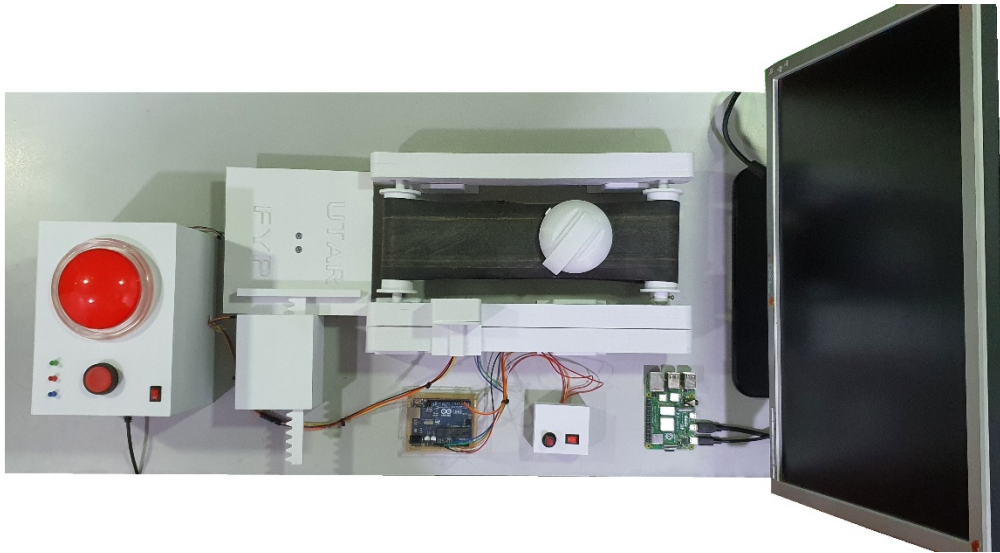


Figure 4.10: Top View of Prototype



Figure 4.11: Front View of Prototype

#### 4.2.1 Mechanical Fabrication

Most of the mechanical parts in this prototype are fabricated by using 3D printing. Firstly, they were designed in 3D modelling software, SolidWorks and saved as stl. file. Next, they were imported into slicing software, Cura to generate g-code for 3D printing. Lastly, g-code was saved in a SD card and executed by a 3D printer. A lot of works like tuning of 3D printing parameters and calibration of bed level are essential to produce a good print. Figure 4.12, Figure 4.13 and Figure 4.14 show the prototype in sections.

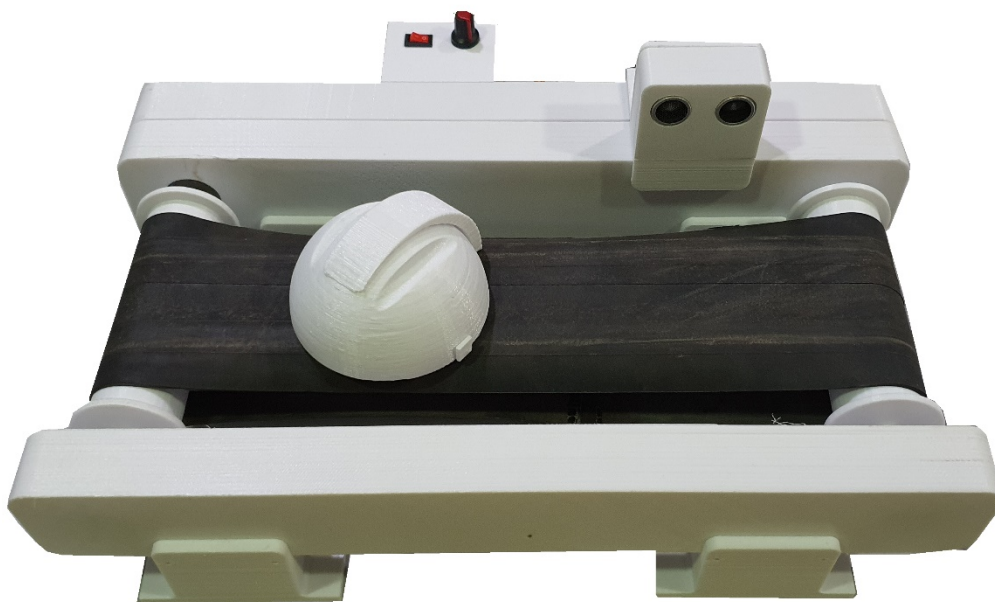


Figure 4.12: Conveyor Belt System

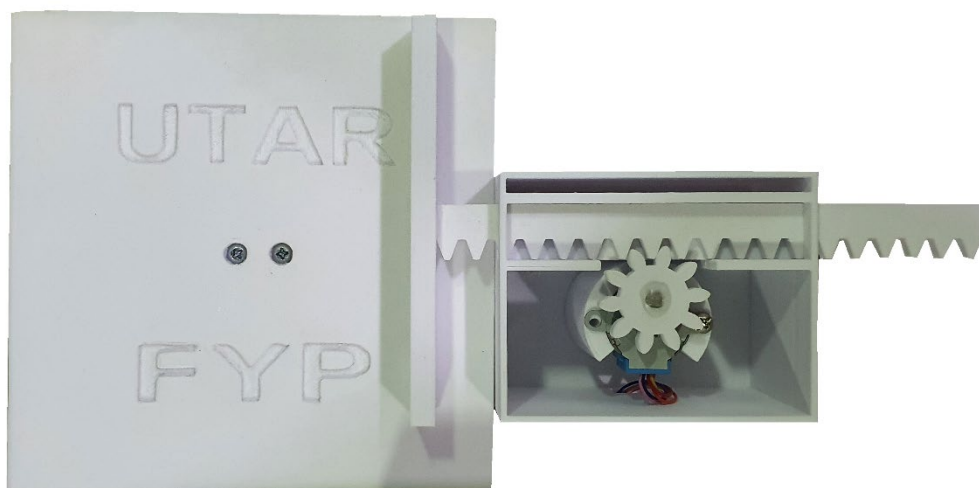


Figure 4.13: Weight Inspection Module with Rejection Mechanism





Figure 4.14: Control Box

#### 4.2.2 Electrical Connection

Electrical circuit design proposed in Chapter 3 was first constructed on a breadboard for testing as shown in Figure 4.15. After everything works fine, they were moved and soldered onto a donut board to reduce the number of wires. Figure 4.16 shows the electrical circuit constructed on donut board.

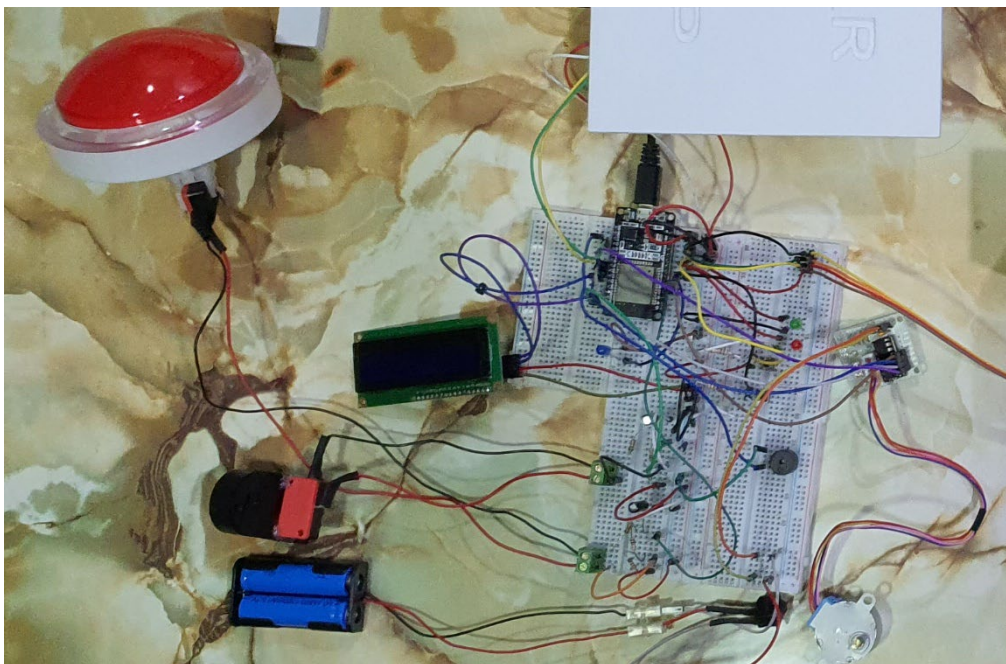


Figure 4.15: Electrical Circuit Constructed on Breadboard

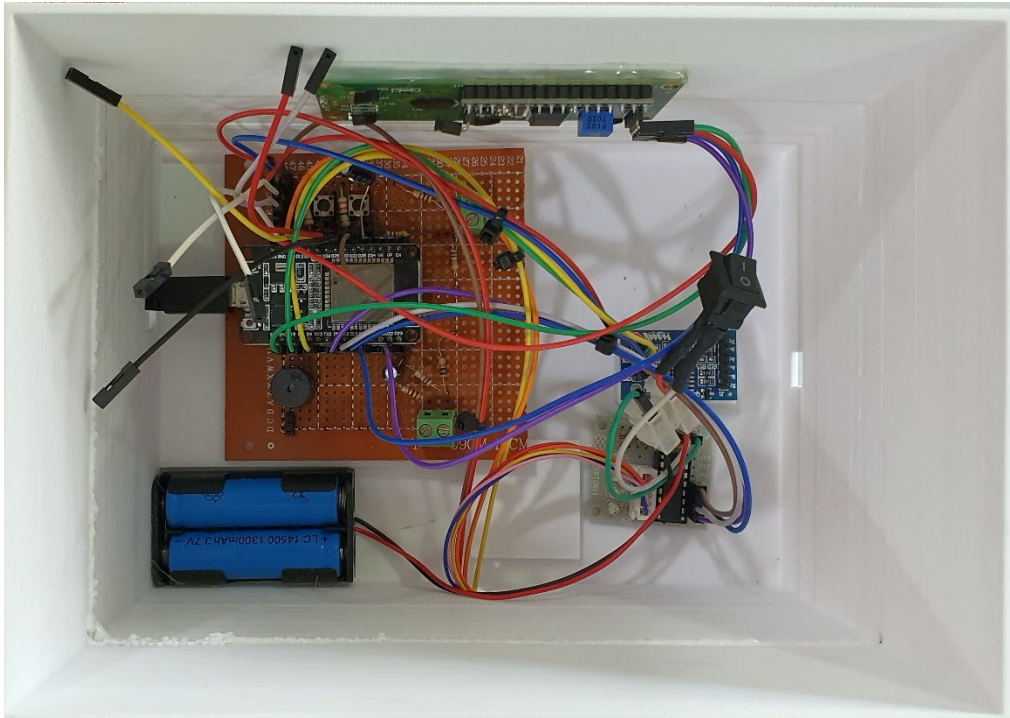


Figure 4.16: Electrical Circuit Constructed on Donut Board

### 4.3 Review of Proposed Solution

Figure 4.17 shows the available features offered by proposed solution. Upon completion, the proposed solution is able to:

- (i) Collect production data automatically and upload them to IoT server in real time.
- (ii) Perform OEE analysis on the collected data in cloud.
- (iii) Precise weight control on every safety helmet produced.
- (iv) Send a call alert to manager and engineer when OEE score falls below than 40%.
- (v) Generate OEE daily report in email.
- (vi) Display real-time OEE data on smart phone.
- (vii) Display production information and maintenance schedule on monitor at production site.

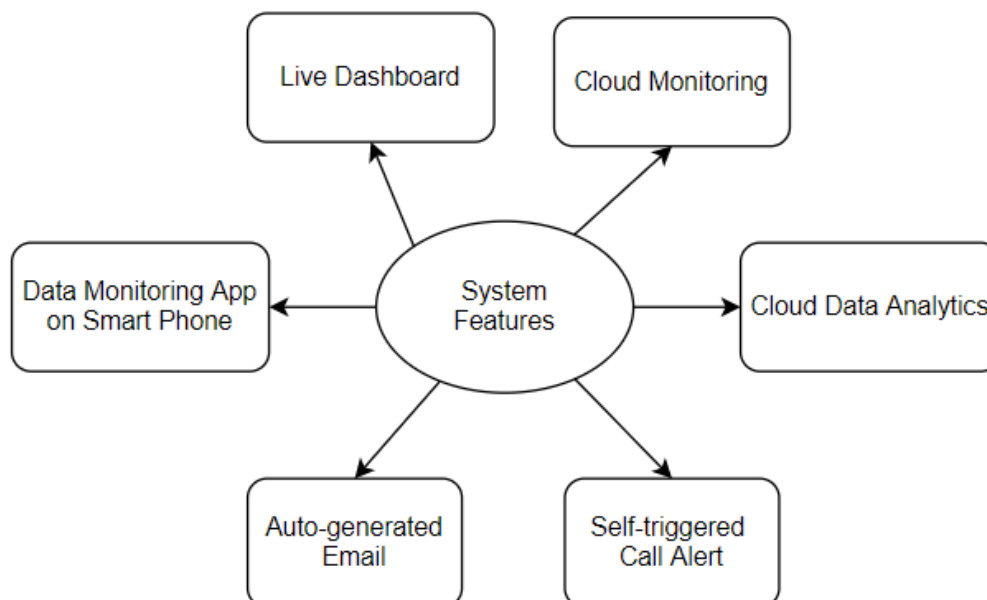


Figure 4.17: Available Features in Proposed Solution

#### 4.3.1 Cloud Monitoring and Data Analytics

This solution is equipped with IoT capability. Hence, it is able to upload collected production data to cloud in real time for remote monitoring. Data points with timestamp will be visualised into graphs to ease the task of monitoring. For instance, engineer can easily spot a downtime event in the graph of total number of units produced if the distance between two data points is too far away. Besides this, rejection of defective products can be easily observed from the graph of rejected amount because it will show an increment step when rejection occurs. Therefore, these graphs can help engineers to quickly discover problem or sense abnormality during production. Figure 4.18, Figure 4.19 and Figure 4.20 show the graphs of total number of units produced, rejected amount and total downtime respectively. Managers and engineers can view these graphs by logging into their ThingSpeak channel.

In addition, a Matlab program is written to perform OEE analysis on the collected data in ThingSpeak. Figure 4.21, Figure 4.22, Figure 4.23 and Figure 4.24 show the analysis results in terms of availability index, performance index, quality index as well as OEE score. With the knowledge of OEE and TPM, engineers can identify the major sources of production loss by looking at the analysis results.

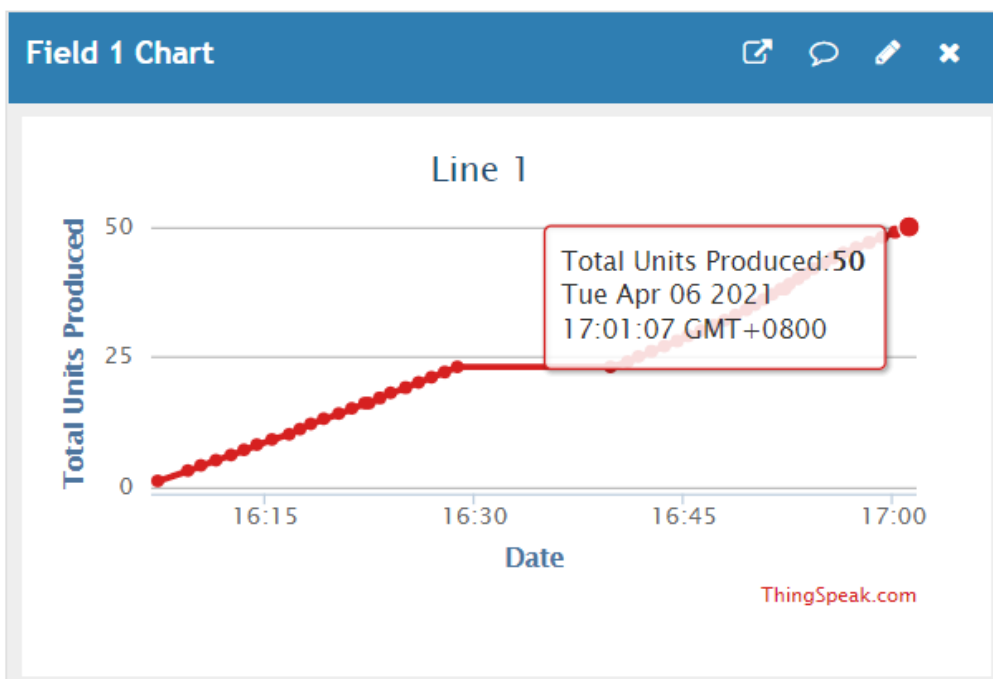


Figure 4.18: Graph of Total Number of Units Produced against Time

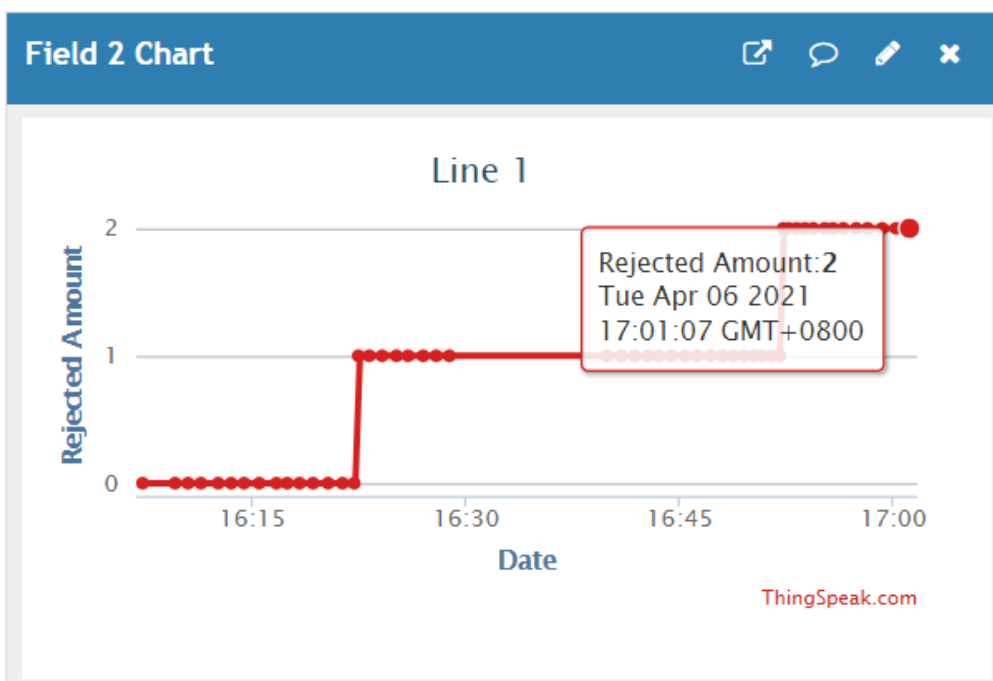


Figure 4.19: Graph of Rejected Amount against Time

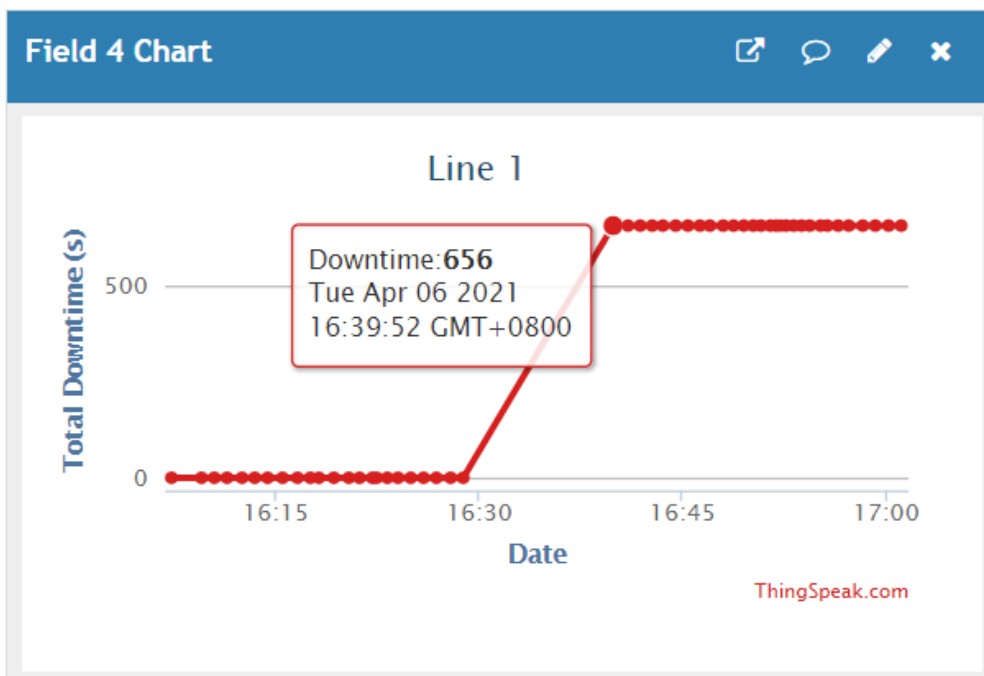


Figure 4.20: Graph of Total Downtime against Time

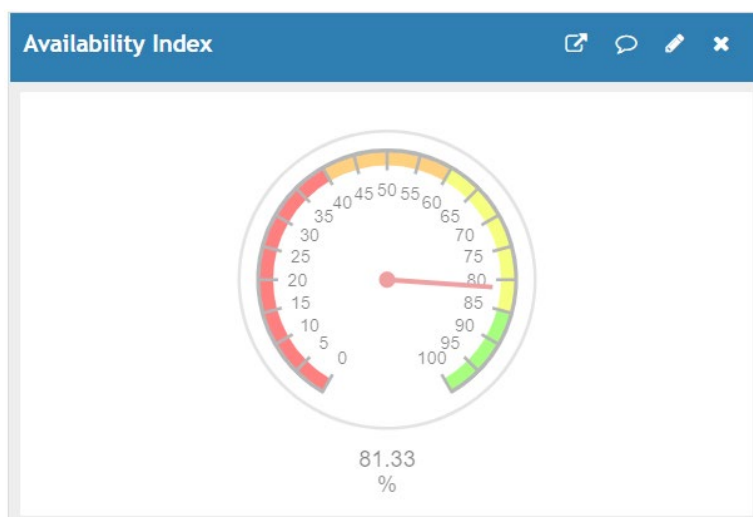


Figure 4.21: Availability Index

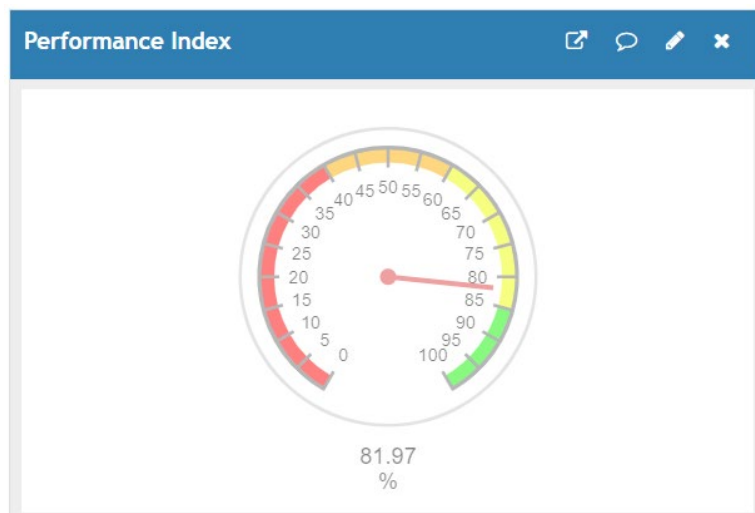


Figure 4.22: Performance Index

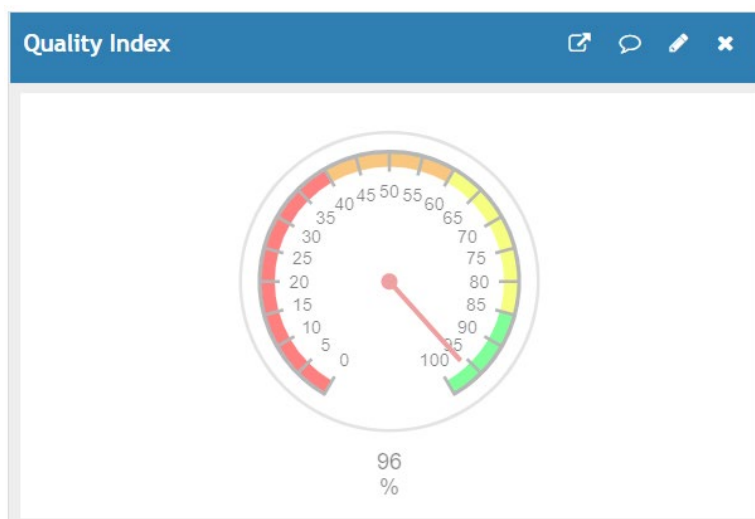


Figure 4.23: Quality Index

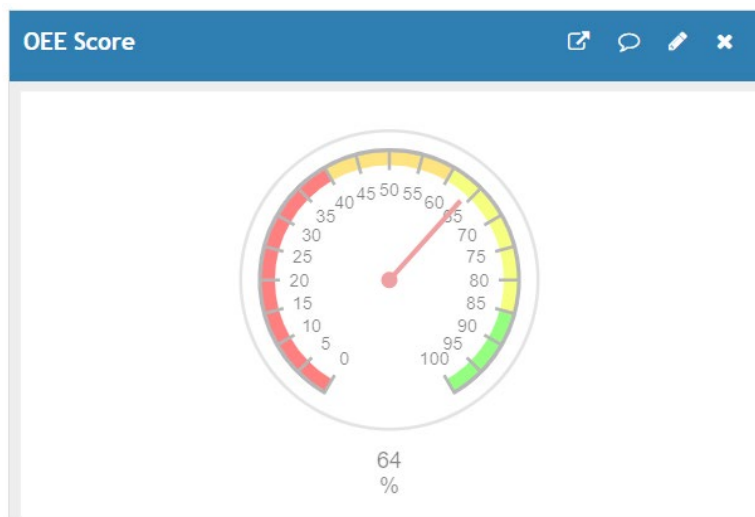


Figure 4.24: OEE Score

### 4.3.2 Self-triggered Call Alert

A self-triggered call alert was configured on IFTTT platform as shown in Figure 4.25. It is programmed to call managers or person in-charge when OEE score falls below than 40%. When problem occurs, this feature can inform managers in the first place so that appropriate action can be taken to avoid subsequent losses. It is an important feature because some operators tend to sweep problem under the carpet. Figure 4.26 shows an incoming VoIP call alert.

**Applet ran**  
Apr 06 - 4:01 PM

**If Maker Event "downtime\_alert", then call my device**

The event named "downtime\_alert" occurred on the Maker Webhooks service

**Webhooks** ▼  
Receive a web request  
● Trigger ran, 4:01 PM

**VoIP Calls** ▲  
Call my device  
● Action ran, 4:01 PM

---

**message**

Hello! Please be alert that now OEE Score for Line 1 falls below than 40%.

Figure 4.25: Configuration of Self-triggered Call Alert on IFTTT Platform

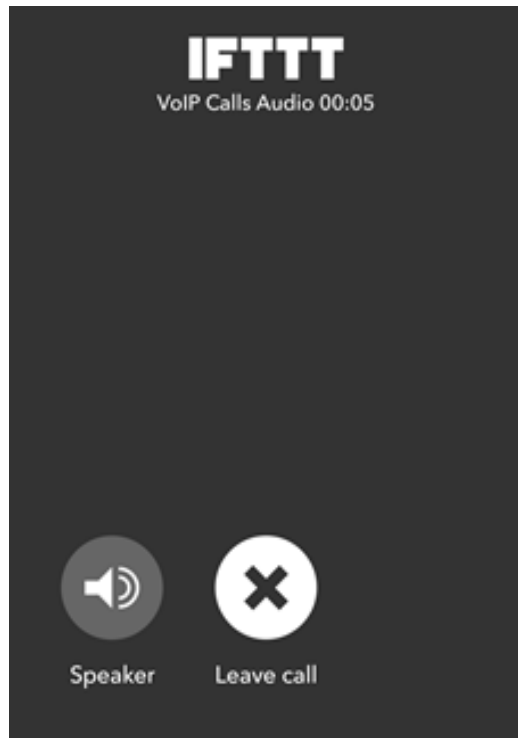


Figure 4.26: Incoming VoIP Call Alert

### 4.3.3 Auto-generated Email of OEE Report

Feature of generating email automatically was configured on IFTTT platform as shown in Figure 4.27. OEE report will be emailed to managers and engineers daily at 12 a.m.. In the auto-generated email, it will provide a quick summary of OEE performance on that day.

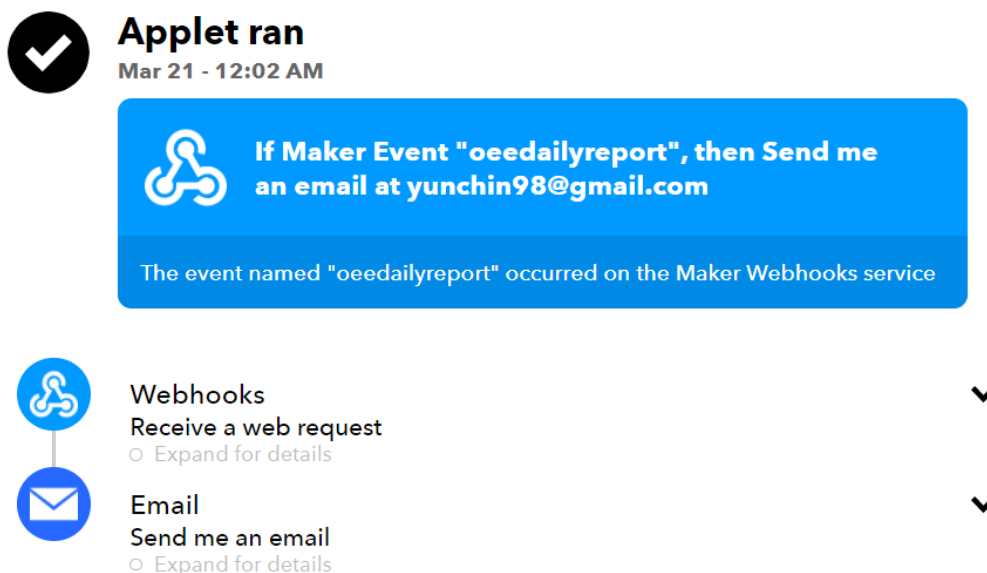


Figure 4.27: Configuration of Auto-generated Email on IFTTT Platform



### 4.3.4 Data Monitoring App on Smart Phone

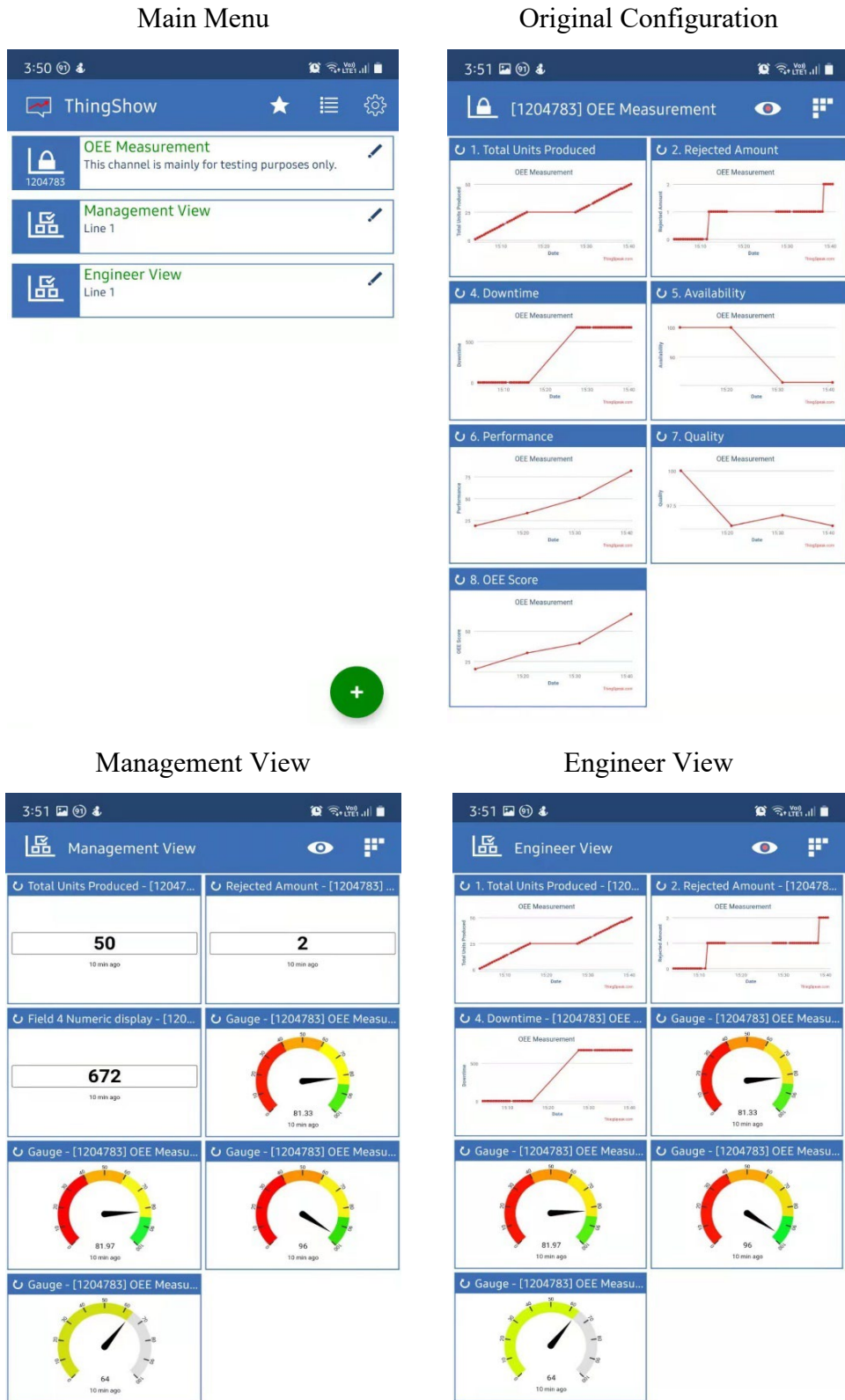


Figure 4.28: Data Monitoring App, ThingShow on Smart Phone

ThingShow is a freeware developed by ThingSpeak and it is available to download from Google Play Store. It can be connected to a private ThingSpeak channel by providing channel ID and API key. In ThingShow, view layout can be configured according to personal preferences. Figure 4.28 shows one original layout and two customize layouts for management and engineers. By using ThingShow, managers and engineers can monitor the status of production at anytime anywhere.

#### 4.3.5 OEE Live Dashboard at Production Site

This live dashboard is intended to replace whiteboard in displaying some important information like production schedule and maintenance schedule. Figure 4.29 shows the main page of live dashboard. Besides, it can be used to display OEE scores so that engineers and operators can know about the current production efficiency. Figure 4.30 depicts the OEE live dashboard for Line 1.

PROGUARD TECHNOLOGIES (M) SDN BHD						
PRODUCTION SCHEDULE						
Line No	Product Code	Mould No	Target Quantity	Hours	Progress	
Line 1	HG-1 (Orange)	MD-456	120	1.5		
Line 2	HG-2 (Red)	MD-486	220	3		
Line 3	HHV 9	MD-486	100	1.5		
Line 4	ALPIN PLUS	MD-726	2000	28		
Line 5	MOMENTUM PRO	MD-488	200	3		
Line 6	ADVANLITE 2	MD-486	230	3		
Line 2	ADV 3 NEW	MD-115	250	3.5		

MAINTENANCE SCHEDULE					
Maintenance Task	Line No	Hours	PIC		Progress
Filter Cleaning	Line 1	1	Ahmad bin Khairul		
Refueling of Hydraulic Oil	Line 2	0.5	Yong Seng Keat		

Figure 4.29: Main Page of Live Dashboard

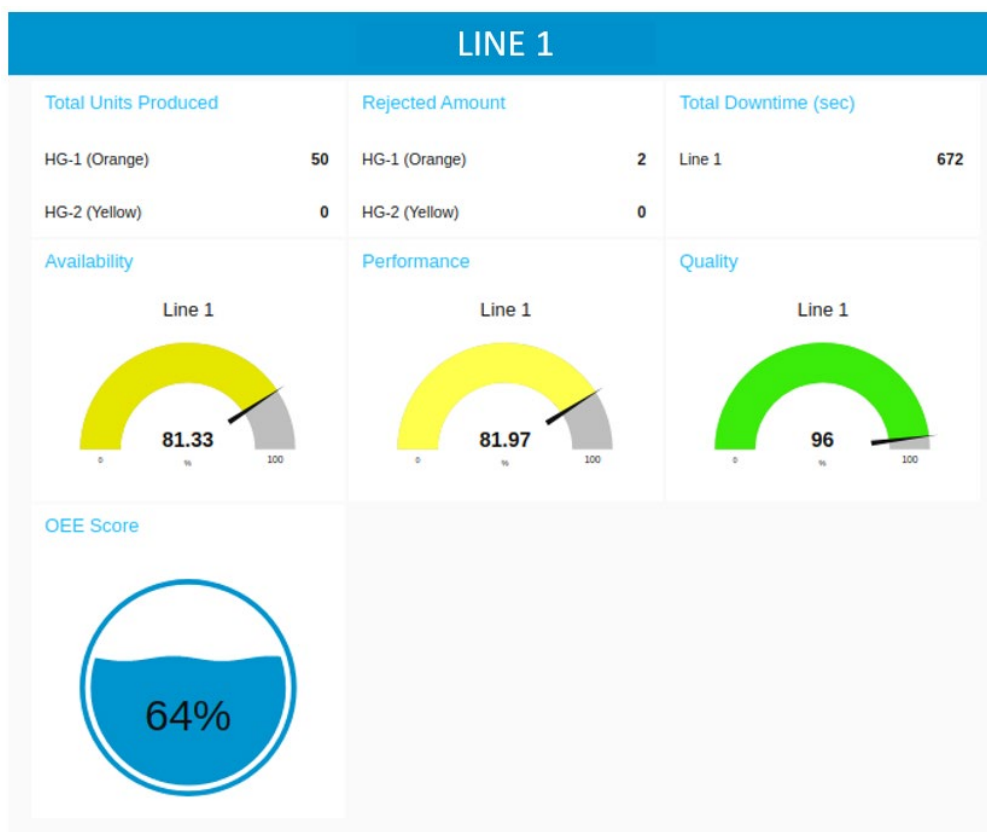


Figure 4.30: OEE Live Dashboard for Line 1

#### 4.4 Challenges and Solutions

Several challenges were encountered during the fabrication of prototype. They were documented under this section along with solutions.

##### 4.4.1 Measurement Noises in Load Cell Sensor

Measurement noises are unavoidable while reading a sensor's values. Two approaches were taken to minimize noises induced during load cell measurement. Firstly, Kalman filter was applied to reduce fluctuations in sensor's reading. It works by predicting the presence of noises and estimating the magnitude of noises via weighted average method. Input readings that are highly repetitive will be assumed as true values and given more weight in the filter model. Table 4.1 shows the comparison between before and after applying Kalman filter in load cell output readings. Noise fluctuation has been reduced around 50% with Kalman filter. Figure 4.31 and Figure 4.32 show the graphs of load cell values against time before and after applying Kalman filter.

Besides this, load cell sensor was programmed to record three measurement readings for every safety helmet. Average value of them will be taken to minimize measurement errors.

Table 4.1: Comparison between Before and After Applying Kalman Filter

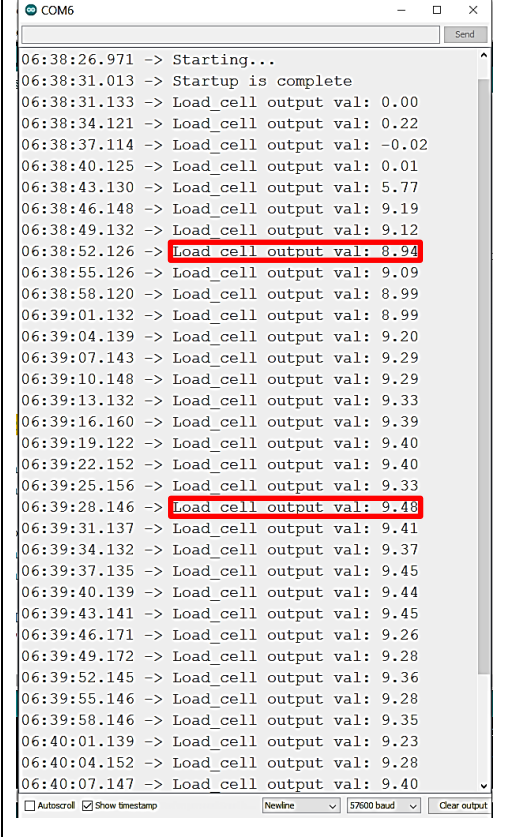
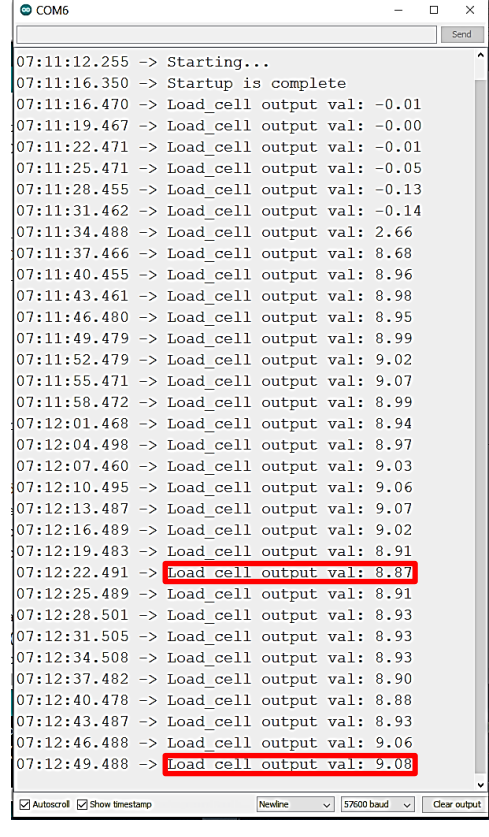
 <pre> 06:38:26.971 -&gt; Starting... 06:38:31.013 -&gt; Startup is complete 06:38:31.133 -&gt; Load_cell output val: 0.00 06:38:34.121 -&gt; Load_cell output val: 0.22 06:38:37.114 -&gt; Load_cell output val: -0.02 06:38:40.125 -&gt; Load_cell output val: 0.01 06:38:43.130 -&gt; Load_cell output val: 5.77 06:38:46.148 -&gt; Load_cell output val: 9.19 06:38:49.132 -&gt; Load_cell output val: 9.12 06:38:52.126 -&gt; Load_cell output val: 8.94 06:38:55.126 -&gt; Load_cell output val: 9.09 06:38:58.120 -&gt; Load_cell output val: 8.99 06:39:01.132 -&gt; Load_cell output val: 8.99 06:39:04.139 -&gt; Load_cell output val: 9.20 06:39:07.143 -&gt; Load_cell output val: 9.29 06:39:10.148 -&gt; Load_cell output val: 9.29 06:39:13.132 -&gt; Load_cell output val: 9.33 06:39:16.160 -&gt; Load_cell output val: 9.39 06:39:19.122 -&gt; Load_cell output val: 9.40 06:39:22.152 -&gt; Load_cell output val: 9.40 06:39:25.156 -&gt; Load_cell output val: 9.33 06:39:28.146 -&gt; Load_cell output val: 9.48 06:39:31.137 -&gt; Load_cell output val: 9.41 06:39:34.132 -&gt; Load_cell output val: 9.37 06:39:37.135 -&gt; Load_cell output val: 9.45 06:39:40.139 -&gt; Load_cell output val: 9.44 06:39:43.141 -&gt; Load_cell output val: 9.45 06:39:46.171 -&gt; Load_cell output val: 9.26 06:39:49.172 -&gt; Load_cell output val: 9.28 06:39:52.145 -&gt; Load_cell output val: 9.36 06:39:55.146 -&gt; Load_cell output val: 9.28 06:39:58.146 -&gt; Load_cell output val: 9.35 06:40:01.139 -&gt; Load_cell output val: 9.23 06:40:04.152 -&gt; Load_cell output val: 9.28 06:40:07.147 -&gt; Load_cell output val: 9.40 </pre>	 <pre> 07:11:12.255 -&gt; Starting... 07:11:16.350 -&gt; Startup is complete 07:11:16.470 -&gt; Load_cell output val: -0.01 07:11:19.467 -&gt; Load_cell output val: -0.00 07:11:22.471 -&gt; Load_cell output val: -0.01 07:11:25.471 -&gt; Load_cell output val: -0.05 07:11:28.455 -&gt; Load_cell output val: -0.13 07:11:31.462 -&gt; Load_cell output val: -0.14 07:11:34.488 -&gt; Load_cell output val: 2.66 07:11:37.466 -&gt; Load_cell output val: 8.68 07:11:40.455 -&gt; Load_cell output val: 8.96 07:11:43.461 -&gt; Load_cell output val: 8.98 07:11:46.480 -&gt; Load_cell output val: 8.95 07:11:49.479 -&gt; Load_cell output val: 8.99 07:11:52.479 -&gt; Load_cell output val: 9.02 07:11:55.471 -&gt; Load_cell output val: 9.07 07:11:58.472 -&gt; Load_cell output val: 8.99 07:12:01.468 -&gt; Load_cell output val: 8.94 07:12:04.498 -&gt; Load_cell output val: 8.97 07:12:07.460 -&gt; Load_cell output val: 9.03 07:12:10.495 -&gt; Load_cell output val: 9.06 07:12:13.487 -&gt; Load_cell output val: 9.07 07:12:16.489 -&gt; Load_cell output val: 9.02 07:12:19.483 -&gt; Load_cell output val: 8.91 07:12:22.491 -&gt; Load_cell output val: 8.87 07:12:25.489 -&gt; Load_cell output val: 8.91 07:12:28.501 -&gt; Load_cell output val: 8.93 07:12:31.505 -&gt; Load_cell output val: 8.93 07:12:34.508 -&gt; Load_cell output val: 8.93 07:12:37.482 -&gt; Load_cell output val: 8.90 07:12:40.478 -&gt; Load_cell output val: 8.88 07:12:43.487 -&gt; Load_cell output val: 8.93 07:12:46.488 -&gt; Load_cell output val: 9.06 07:12:49.488 -&gt; Load_cell output val: 9.08 </pre>
<p>Max: 9.48g Min: 8.94g Fluctuation: <math>\pm 0.54</math>g</p>	<p>Max: 9.08g Min: 8.87g Fluctuation: <math>\pm 0.21</math>g</p>
<p>BEFORE</p>	<p>AFTER</p>



Figure 4.31: Graph of Load Cell Values against Time before Applying Kalman Filter

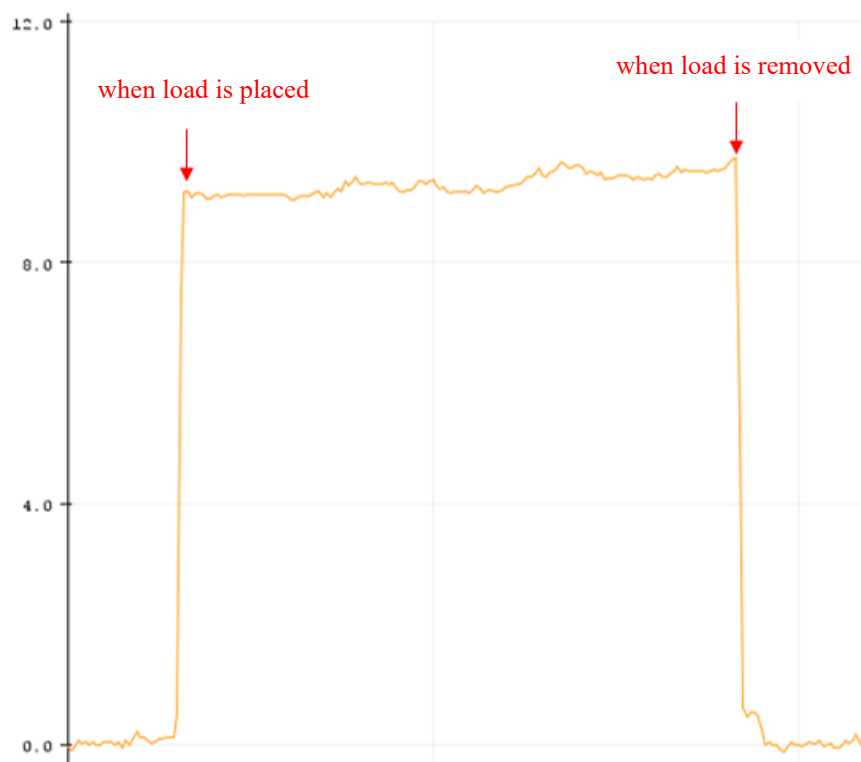


Figure 4.32: Graph of Load Cell Values against Time after Applying Kalman Filter

#### 4.4.2 Switch Bouncing Issues for Rejection and Downtime Push Button

Switch bouncing is a common problem faced when reading an input from switches. When a button is pressed, it might seem like switch is closed instantly. However, in reality, switch will bounce between the states of open and close at the instant of pushing the button. As a result, ESP32 will think that the button is pressed for several times because it has a higher working frequency than switch bounce. Figure 4.33 shows the switch bouncing effect of a switch.

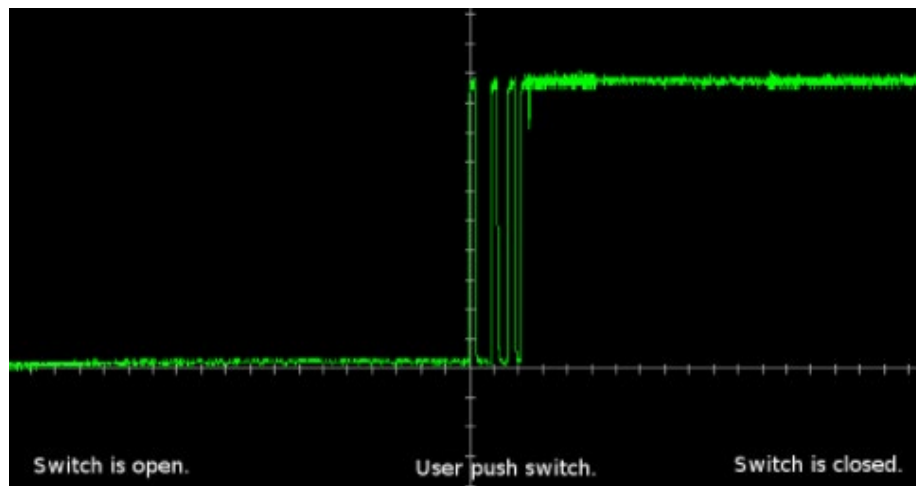


Figure 4.33: Switch Bouncing Effect (Christoffersen, 2015)

This problem can be solved by both hardware and software approaches. By hardware approach, a simple RC switch debouncing circuit was implemented to filter out noises induced when button is pressed. Figure 4.34 shows the schematic diagram of switch debouncing circuit used in this project.

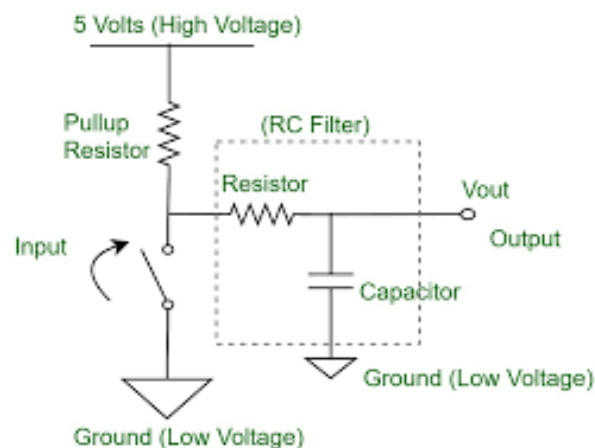


Figure 4.34: Switch Debouncing Circuit (SujayPosam, 2019)

By software approach, it was programmed so that the designated GPIO pins on ESP32 will only accept one input pulse signal for a period of time. Figure 4.35 shows the code written to tell ESP32 to consider only one input pulse for rejection counter in a period of 5 seconds. If multiple input pulses are read within 5 seconds, program will quit the interrupt event without doing anything.

```
//Interrupt Event
void isr(){
    interrupt_time = millis();

    if (interrupt_time - last_interrupt_time > 5000)
    {
        rejectedcount = rejectedcount + 1;
        last_interrupt_time = interrupt_time;
    }
}
```

Figure 4.35: Software Approach to Overcome Switch Bouncing Effect

#### 4.4.3 NEMA 17 Stepper Motor Vibration Issue

It is observed that NEMA 17 stepper motor vibrates vigorously and produces a lot of noises when it is working in full step mode. In full step mode, NEMA 17 stepper motor has only 200 steps for one revolution. When NEMA 17 stepper motor operates at a relatively low speed, it will produce a very “rough” motion due to a large step angle. Other than using damper to reduce motor vibration, microstepping is another alternative to smoothen motor rotation by reducing step size. A4988 stepper motor driver allows microstepping up to sixteenth step resolution. Table 4.2 depicts the pin configurations for microstepping in A4988 stepper motor driver. In this project, half step resolution was applied to stepper motor for conveyor belt system by connecting 3.3V pin on Arduino UNO to MS1 pin on A4988. The reasons of selecting half step resolution are half step is sufficient to produce a very smooth motion and higher microstepping resolution may sacrifice the incremental torque of stepper motor.

Table 4.2: Microstepping Configuration in A4988 Stepper Motor Driver Board

MS1	MS2	MS3	Resolution
LOW	LOW	LOW	Full Step
HIGH	LOW	LOW	Half Step
LOW	HIGH	LOW	Quarter Step
HIGH	HIGH	LOW	Eighth step
HIGH	HIGH	HIGH	Sixteenth Step



## CHAPTER 5

### CONCLUSION AND RECOMMENDATION

#### 5.1 Conclusion

This project is aimed to implement OEE on safety helmet plastic injection process at Proguard Technologies (M) Sdn Bhd. Two objectives to be achieved are to measure machine efficiency by performing OEE analysis and to reduce 80% time spent for data collection process by developing an automatic data acquisition system with IoT capability. A Microsoft Excel template was created to perform OEE analysis. Based on the data collected on 2<sup>nd</sup> of April 2021, Proguard Technologies (M) Sdn Bhd achieved an OEE score of 63.89% for their production line. According to OEE Score World Rankings, Proguard Technologies (M) Sdn Bhd is categorized as typical manufacturers. In line with second objective, a solution of automatic data acquisition system with IoT capability was proposed. Initially, data collection system adopted by Proguard Technologies (M) Sdn Bhd is record by using paper and insert manually into computer. 15 minutes will be spent to collect data before every shift ends. There are two shifts in one day, so 30 minutes are spent to collect data everyday. By implementing the proposed solution, time spent for data collection process can be saved up to 100%. Table 5.1 shows the comparison between before and after applying solution proposed in this project. To wrap it up, both objectives of this project were achieved.

Table 5.3: Comparison between Before and After Applying Proposed Solution

Time Spent Per Day for Data Collection Process	
BEFORE	AFTER
30 Mins	0 Mins

#### 5.2 Limitation and Recommendation

Several limitations are observed in this project. Firstly, the proposed data acquisition system cannot upload data to its IoT server in every second. This is because the shortest time interval between two data points allowed by

ThingSpeak server are 15 seconds. It is still acceptable in this project because the cycle time of safety helmet plastic injection process is much more than 15 seconds. However, it is recommended to use Ubidots server instead of ThingSpeak server if data is required to be updated in every second. Despite this, Ubidots in free version does not support data analytics like ThingSpeak.

Another limitation of this project is the level of automation. Push buttons were used to collect the data of rejection and downtime. It requires human labour to trigger the push buttons. It is recommended to implement machine vision for rejection due to plastic defects in future. Moreover, AI algorithm can be used to track and predict unplanned downtime event like equipment breakdown based on the historical failure rate of the machine.

## REFERENCES

- Ahuja, I.P.S. and Khamba, J.S., 2008. Total productive maintenance: literature review and directions. *International Journal of Quality and Reliability Management*, 25(7), pp. 709-756.
- Andersson, C. and Bellgran, 2011. Combining Overall Equipment Efficiency (OEE) and productivity measures as drivers for production improvements. *Swedish Production Symposium 2011*
- Andrew, M., 2001. *Plastic Injection Moulding – An Introduction*. The Institute of Materials. [online] Available at: <<https://www.azom.com/article.aspx?ArticleID=265>> [Accessed 5 September 2020]
- Christoffersen, J., 2015. *Switch Bounce and How to Deal with It*. [online] Available at: <<https://www.allaboutcircuits.com/technical-articles/switch-bounce-how-to-deal-with-it/>> [Accessed 10 April 2021].
- Ellacott, J., 2019. *5 causes of equipment failure (and what you can do to prevent it)*. [online] Available at: <<https://www.fixsoftware.com/blog/5-causes-of-equipment-failure-and-what-you-can-do-to-prevent-it/>> [Accessed 4 September 2020].
- Fleischer, J., Weismann, U. and Niggeschmidt, S., 2006. Calculation and optimization model for costs and effects of availability relevant service elements. *Proceedings of LCE2006*.
- Giegling, S., Haymon, T., Dr. William, A.V. and Dr. Konopka, J., 1997. Implementation of Overall Equipment Effectiveness at a Semiconductor Manufacturer. *Electronics Manufacturing Technology Symposium*
- Gupta, A.K. and Garg, R.K., 2012. OEE Improvement by TPM Implementation: A Case Study. *International Journal of IT, Engineering and Applied Sciences Research*, 1(1), pp. 115-124.

Huang, S.H., Dismukes, J.P., Mousalam, A., Razzak, R.B. and Robinson, D.E., 2003. Manufacturing productivity improvement using effectiveness metrics and simulation analysis. *Internal Journal of Production Response*, 41, pp. 513-527.

Jain, A., Bhatti, R.S. and Singh, H., 2015. OEE enhancement in SMEs through mobile maintenance: a TPM concept. *International Journal of Quality & Reliability Management*, [online] 32(5), pp. 503-516. Available at: <<https://doi.org/10.1108/IJQRM-05-2013-0088>>

Jonsson, P. and Lesshammar, M., 1999. Evaluation and improvement of manufacturing performance measurement systems – the role of OEE. *International Journal of Operations & Production Management*, [online] 19(1), pp. 55-78. Available at: <<https://doi.org/10.1108/01443579910244223>>

Keegan, M.C., n.d. Analysis of Overall Equipment Effectiveness (OEE) within different sectors in different Swedish industries.

Kumar, J. and Soni, V.K., 2014. An exploratory study of OEE implementation in Indian manufacturing companies. *Journal of The Institution of Engineers (India)*, [online]. Available at: <<https://doi.org/10.1007/s40032-014-0153-x>>

Liliane, M.A.P. and Peter, N.M., 2008. Performance measurement using overall equipment effectiveness (OEE): Literature review and practical application discussion. *International Journal of Production Research*, 46 (13), pp. 3517-3535.

Ljungberg, 1998. Measurement of overall equipment effectiveness as a basis for TPM activities. *International Journal of Operations and Production Management*, 18(5), pp. 495-507.

Ljungberg, O., 1998. Measurement of overall equipment effectiveness as a basis for TPM activities. *International Journal of Operations & Production Management*, [online] 18(5), pp. 495-507. Available at: <<http://dx.doi.org/10.1108/01443579810206334>>

Maharjan, S., 2011. *Implementing the 5S Methodology for the Graphic Communications Management*. [online] Available at: <<http://www2.uwstout.edu/content/lib/thesis/2011/2011maharjans.pdf>>

Michalska, J. and Szewieczek, D., 2007. The 5S methodology as a tool for improving the organisation. *Journal of Achievements in Materials and Manufacturing Engineering*, 24(2), pp. 211-214.

Mila, B., 2018. *What are the Six Big Losses in manufacturing and how to eliminate them*. [online] Available at: <<https://www.zaptic.com/insights/what-are-the-six-big-losses-in-manufacturing>> [Accessed 4 September 2020].

Muchiri, P. and Pintelon, L., 2006. Performance measurement using overall equipment effectiveness (OEE): literature review and practical application discussion. *International Journal of Production Research*, [online] 46(13), pp. 3517-3535. Available at: <<http://dx.doi.org/10.1080/00207540601142645>>

Nakajima, S., 1988. *TPM Development Program: Implementing Total Productive Maintenance*.

Padilla and Pekmezci, 2011. *A Framework for Designing a Lean Production System for SMEs*. [online] Available at: <<https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A424709&dswid=-5435>>

Peter, T., 2018. *Why the Six Big Losses are crucial to your preventative maintenance regime*. [online] Available at: <<https://www.sageautomation.com/blog/why-the-six-big-losses-are-crucial-to-your-preventative-maintenance-regime>> [Accessed 3 September 2020].

Santiago, G.G., n.d. *What is TPM?* [online] Available at: <<http://www.mantenimientopetroquimica.com/en/predictivemaintenance/144-articles-of-interest/95-what-is-tpm.html>> [Accessed 4 September 2020].

Singh, R.K., Clements, E.J. and Sonwaney, V., 2018. Measurement of overall equipment effectiveness to improve operational efficiency. *International Journal of Process Management and Benchmarking*, [online] 8(2). Available at: <<http://dx.doi.org/10.1504/IJPMB.2018.10010267>>

SujayPosam, 2019. *Switch Debounce in Digital Circuits*. [online] Available at: <<https://www.geeksforgeeks.org/switch-debounce-in-digital-circuits/>> [Accessed 10 April 2021].

Thirumalvalavan, K. and Ananth, A.A., 2017. Quality of work life and job satisfaction of employees in steel authority of India Ltd, Salem. *International Journal of Academic Research and Development*, 2(6), pp. 723-727.

Venkatesh, J., 2005. *An Introduction to Total Productive Maintenance (TPM)*. The Plant Maintenance Resource Center. [online] Available at: <<http://faculty.nps.edu/dl/sysengineering/se3302/pdf/anintroductiontototalproductivemaintenance.pdf>> [Accessed 1 September 2020]

## APPENDICES

### APPENDIX A: Main Program in ESP32 Microcontroller

```
//Include library
#include <HX711_ADC.h>
#include <LiquidCrystal_I2C.h>
#include <Stepper.h>
#include <WiFi.h>
#include <Wire.h>
#include "ThingSpeak.h"
#include "time.h"

//Define constants
LiquidCrystal_I2C lcd(0x27, 16, 2);
Stepper stepper(2038, 5, 19, 18, 32);

WiFiClient client;
char ssid[] = "kexinchua95@unifi";
char pass[] = "01311100";
int status = WL_IDLE_STATUS;

unsigned long myChannelNumber = 1204783;
const char * myWriteAPIKey = "AXG6LB4XT9OFKDAY";

//Ultrasonic Sensor
const int trigPin = 12;
const int echoPin = 13;

//ESP32 Deep Sleep Mode
const int wakeup_button = 33;
const int sleep_button = 14;
```

```
const int sleep_led = 26;
//Wifi Connection Indicator
const int wifi_status = 27;

//Push Button for Rejection
const int rej_button = 35;

//Good Product Indicator
const int good_led = 25;

//Push Button for Downtime
const int downtime_button = 23;

//NTP Time Server
const char * ntpServer = "pool.ntp.org";
const long gmtOffset_sec = 25200;
const int daylightOffset_sec = 3600;

//Variables
long duration;
int distance;
int sleepmode_status;
int totalUnitsProduced = 0;
int last_totalUnitsProduced = 0;
float weight = 0;
float last_weight = 0;
float i;
float weight1;
float weight2;
float weight3;
long t;
```



```
//Interrupt Variables
volatile int rejectedcount = 0;
volatile int last_rejectedcount = 0;
volatile int interrupt_time;
volatile int interrupt_time1;
volatile int last_interrupt_time = 0;
volatile int last_interrupt_time1 = 0;
volatile unsigned long StartTime;
volatile unsigned long EndTime;
volatile long ElapsedTime = 0;
volatile long TotalDowntime = 0;
volatile long LastTotalDowntime = 0;
volatile int iteration = 0;

//Load Cell Pins
const int HX711_dout = 4;
const int HX711_sck = 2;

//HX711 constructor:
HX711_ADC LoadCell(HX711_dout, HX711_sck);

//Interrupt Event
void isr(){
    interrupt_time = millis();

    if (interrupt_time - last_interrupt_time > 5000)
    {
        rejectedcount = rejectedcount + 1;
        last_interrupt_time = interrupt_time;
    }
}
```

```
//Downtime Interrupt Event
void isr2(){
  interrupt_time1 = millis();
  if (interrupt_time1 - last_interrupt_time1 > 10000)
  {
    iteration++;

    if (iteration % 2 == 0)
    {
      EndTime = millis();
      ElapsedTime = EndTime - StartTime;
      ElapsedTime = ElapsedTime / 1000;
      TotalDowntime = TotalDowntime + ElapsedTime;

      Serial.print("DownTime: ");
      Serial.print(ElapsedTime);
      Serial.println("s");
      digitalWrite(26,LOW);
    }else
    {
      StartTime = millis();
      digitalWrite(26,HIGH);
    }

    last_interrupt_time1 = interrupt_time1;
  }
}

void setup()
{
  Serial.begin(115200);
```

```
delay(10);

//Initialize the LCD,
lcd.begin();
lcd.backlight();
Serial.println("Connecting to ");
Serial.println(ssid);
WiFi.begin(ssid, pass);

while (WiFi.status() != WL_CONNECTED)
{
    delay(500);
    Serial.print(".");
}

Serial.println("");
Serial.println("WiFi connected");

//Wifi Connection Indicator
pinMode(wifi_status,OUTPUT);
digitalWrite(wifi_status,HIGH);

//Init and get the time
configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);
printLocalTime();

ThingSpeak.begin(client);

//Ultrasonic Sensor
pinMode(trigPin,OUTPUT);
pinMode(echoPin,INPUT);
```

```
//Deep Sleep Mode
pinMode(sleep_led,OUTPUT);
pinMode(sleep_button,INPUT);

//Enable External Wake Up
esp_sleep_enable_ext0_wakeup(GPIO_NUM_33,1);
//Push Button for Rejection
pinMode(rej_button,INPUT);
attachInterrupt(35, isr, FALLING);

//Push Button for Downtime
pinMode(downtime_button,INPUT);
attachInterrupt(23, isr2, FALLING);

//Good Product Indicator
pinMode(good_led,OUTPUT);

//Buzzer setup
ledcSetup(0,1E5,12);
ledcAttachPin(15,0);

LoadCell.begin();
float calibrationValue;
calibrationValue = 422.34;

long stabilizingtime = 2000;
boolean _tare = true;
LoadCell.start(stabilizingtime, _tare);
if (LoadCell.getTareTimeoutFlag()) {
  Serial.println("Timeout, check MCU>HX711 wiring and pin designations");
  while (1);
}
```

```
    }  
    else {  
        LoadCell.setCalFactor(calibrationValue);  
        Serial.println("Setup is complete");  
    }  
}
```

```
void loop()  
{  
    static boolean newDataReady = 0;  
    const int serialPrintInterval = 0;  
  
    digitalWrite(trigPin,LOW);  
    delayMicroseconds(2);  
    digitalWrite(trigPin,HIGH);  
    delayMicroseconds(10);  
    digitalWrite(trigPin,LOW);  
  
    duration = pulseIn(echoPin,HIGH);  
    distance = duration*0.034/2;  
  
    Serial.print("Distance: ");  
    Serial.println(distance);  
  
    printLocalTime();  
  
    lcd.setCursor (0,1);  
  
    if (distance <= 10)  
    {  
        totalUnitsProduced++;
```

```
Serial.print("Total Units Produced: ");
Serial.println(totalUnitsProduced);
lcd.print("TOT:");
lcd.print(totalUnitsProduced);

//For checking purposes
do
{
    digitalWrite(trigPin,LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin,HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin,LOW);

    duration = pulseIn(echoPin,HIGH);
    distance = duration*0.034/2;

    if (distance <= 10)
    {
        Serial.println("Inspection In Progress");
        delay(1000);
    }
    else
    {
        Serial.println("Inspection Done");
    }
}while (distance <= 10);
}
else
{
    lcd.print("TOT:");
```

```
    lcd.print(totalUnitsProduced);
}

Serial.print("Rejected Quantity: ");
Serial.println(rejectedcount);
lcd.print(" REJ:");
lcd.print(rejectedcount);
delay(50);

//Check for new data/start next conversion:
if (LoadCell.update()) newDataReady = true;
//Get smoothed value from the dataset:
if (newDataReady){
    if (millis() > t + serialPrintInterval){
        i = LoadCell.getData();
        Serial.print("Load cell output value: ");
        Serial.println(i);
        newDataReady = 0;
        t = millis();
    }

    if (i >= 200)
    {
        weight1 = i;
        delay(1000);
        LoadCell.update();
        i = LoadCell.getData();
        Serial.println(i);

        if (i >= 200)
        {
```

```
weight2 = i;
delay(1000);
LoadCell.update();
i = LoadCell.getData();
Serial.println(i);

if (i >= 200)
{
    weight3 = i;
    delay(1000);

    weight = (weight1 + weight2 + weight3) / 3 ;

    lcd.clear();
    lcd.print("WEIGHT: ");
    Serial.print("WEIGHT: ");
    lcd.print(weight);
    Serial.print(weight);
    lcd.print("g");
    lcd.noBlink();
    Serial.println("g");

    if (weight < 298 || weight > 302)
    {
        rejectedcount = rejectedcount + 1;
        stepper.setSpeed(15);

        ledcWriteTone(0,1000);
        delay(500);
        ledcWriteTone(0,0);
        delay(500);
```



```
    ledcWriteTone(0,1000);
    delay(500);
    ledcWriteTone(0,0);
    delay(500);
    ledcWriteTone(0,1000);
    delay(500);
    ledcWriteTone(0,0);
    delay(500);
    ledcWriteTone(0,1000);
    delay(500);
    ledcWriteTone(0,0);
    delay(500);
    ledcWriteTone(0,1000);
    delay(500);
    ledcWriteTone(0,0);
    delay(500);

    stepper.step(1250);
    stepper.step(-1250);
}else
{
    digitalWrite(good_led,HIGH);
}

delay(10000);
digitalWrite(good_led,LOW);

}
}
}
```

```

if (totalUnitsProduced != last_totalUnitsProduced || rejectedcount !=
last_rejectedcount || weight != last_weight || TotalDowntime !=
LastTotalDowntime)
{
    last_totalUnitsProduced = totalUnitsProduced;
    last_rejectedcount = rejectedcount;
    last_weight = weight;
    LastTotalDowntime = TotalDowntime;

    ThingSpeak.setField(1,totalUnitsProduced);
    ThingSpeak.setField(2,rejectedcount);
    ThingSpeak.setField(3,weight);
    ThingSpeak.setField(4,TotalDowntime);
    ThingSpeak.writeFields(myChannelNumber,myWriteAPIKey);
    Serial.println("Data Updated");
}

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

//Check if last tare operation is complete:
if (LoadCell.getTareStatus() == true) {
    Serial.println("Tare complete");
}

sleepmode_status = digitalRead(sleep_button);

if (sleepmode_status == HIGH)

```

```
{
  Serial.println("Entering Sleep");
  lcd.clear();
  lcd.print("Entering Sleep");
  digitalWrite(sleep_led,HIGH);
  delay(1000);
  digitalWrite(sleep_led,LOW);
  delay(1000);
  digitalWrite(sleep_led,HIGH);
  delay(1000);
  digitalWrite(sleep_led,LOW);
  delay(1000);
  digitalWrite(sleep_led,HIGH);
  delay(1000);
  digitalWrite(sleep_led,LOW);
  lcd.clear();
  lcd.print("Sleeping Mode");
  esp_deep_sleep_start();
}
}

void printLocalTime(){
  struct tm timeinfo;
  if(!getLocalTime(&timeinfo)){
    Serial.println("Failed to obtain time");
    return;
  }

  lcd.clear();
  lcd.print(&timeinfo, "%d");
  lcd.print(" ");
```

```
lcd.print(&timeinfo, "%B");  
lcd.print(" ");  
lcd.print(&timeinfo, "%Y");  
  
if(timeinfo.tm_hour == 0 && timeinfo.tm_min == 5 && timeinfo.tm_sec == 0)  
{  
    totalUnitsProduced = 0;  
    rejectedcount = 0;  
    weight = 0;  
    TotalDowntime = 0;  
}  
}
```

## APPENDIX B: Matlab Program to Compute OEE Score

```

% Channel ID to read data from
readChannelID = 1204783;

% Field ID
totalunitsproducedFieldID = 1;
rejectedamountFieldID = 2;
downtimeFieldID = 4;

% Channel Read API Key
readAPIKey = 'HHXBL1BKCHHL7S5M';

% Read Data from Channel
totalunitsproduced =
thingSpeakRead(readChannelID, 'Fields', totalunitsproducedF
ieldID, 'NumPoints', 1, 'ReadKey', readAPIKey);
rejectedamount =
thingSpeakRead(readChannelID, 'Fields', rejectedamountField
ID, 'NumPoints', 1, 'ReadKey', readAPIKey);
downtime =
thingSpeakRead(readChannelID, 'Fields', downtimeFieldID, 'Nu
mPoints', 1, 'ReadKey', readAPIKey);

% Calculate Availability Index
tppt = 24*60*60;
apt = tppt - downtime;
availability = apt / tppt * 100;
availability2 = round(availability, 2);
display(availability, 'Availability');
display(availability2, 'Availability');

```

```
% Calculate Performance Index
startTime = 0;
timeStamp = datetime('now');
display(timeStamp, 'TimeStamp');
h = hour(timeStamp);
m = minute(timeStamp);
s0 = second(timeStamp);
s = round(s0);
totaltimeStamp = h*60*60 + m*60 + s
display(totaltimeStamp, 'totaltimeStamp');
elapsedTime = totaltimeStamp - startTime;
display(elapsedTime, 'Elapsed Time');
apt2 = totaltimeStamp - downtime; %elapsedTime
ict = 0.8*60; % Input unit: mins / part
performance = ict * totalunitsproduced * 100 / apt2;
performance2 = round(performance,2);
display(performance, 'Performance');
display(performance2, 'Performance');

% Calculate Quality Index
quality = (totalunitsproduced - rejectedamount) * 100 /
totalunitsproduced;
quality2 = round(quality,2);
display(quality, 'Quality');
display(quality2, 'Quality');

% Calculate OEE Score
oee = availability * performance * quality / 10000;
oee2 = round(oee,2);
display(oee, 'OEE');
display(oee2, 'OEE');
```

```
% Write Data Channel ID
writeChannelID = 1204783;

% Write API Key
writeAPIKey = 'AXG6LB4XT90FKDAY';

% Write Data into Channel
thingSpeakWrite(writeChannelID,[availability2,performance
2,quality2,oe2], 'Fields',[5,6,7,8], 'WriteKey',writeAPIKey);
```

**APPENDIX C: Matlab Program to Reset Parameters Daily**

```
% Field ID
totalunitsproducedFieldID = 1;
rejectedamountFieldID = 2;
downtimeFieldID = 4;

% Declaration
totalunitsproduced = 0;
rejectedamount = 0;
downtime = 0;
availability = 0;
performance = 0;
quality = 0;
oee = 0;

% Write Data Channel ID
writeChannelID = 1204783;

% Write API Key
writeAPIKey = 'AXG6LB4XT90FKDAY';

% Write Data into Channel
thingSpeakWrite(writeChannelID,[availability,performance,
quality,oee], 'Fields',[5,6,7,8], 'WriteKey',writeAPIKey);
```