IOT BASED PALM OIL MILL AUTOMATION IN MONITORING OIL EXTRACTION RATE

JESSICA HON SIAU CHIEH

A project report submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering (Honours) Biomedical 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 : Jessica Hon Siau Chieh

ID No. : 15UEB04165

Date : 16 May 2020

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"IOT BASED PALM OIL MILL AUTOMATION IN MONITORING OIL EXTRACTION RATE"** was prepared by **JESSICA HON SIAU CHIEH** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Engineering (Honours) Biomedical Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature	:	
Supervisor	:	Ts Dr Lin Horng Sheng
Date	:	16/5/2020

Signature	:	
Co-Supervisor	:	Ts Phua Yeong Nan
Date	:	16/5/2020

ACKNOWLEDGEMENTS

I would like to thank everyone whose assistance was a milestone in the completion of this project. I would like to express my sincere appreciation to my research supervisor, Ts Dr Lin Horng Sheng for his insightful advice, guidance and his enormous patience throughout the development of the research. Without his persistent help, the goal of this project would not have been realized.

In addition, I would also like to acknowledge the support and great love of my family and friends who had helped and given me encouragement. They kept me going on and this work would not have been possible without their input.

ABSTRACT

Oil extraction rate (OER) can be used to assess the efficiency or profitability of a palm oil mill, in which it measures the crude palm oil (CPO) extraction output against the fresh fruit bunches (FFB) of oil palm input into the mill. Nowadays, the oil level monitoring system is based on manual way by using a sounding meter to measure the oil in the storage tank. This manual method can cause a less accurate data acquisition in addition to the waste of time and manpower in executing the task. This method also causes loss of oil in factory due to smuggling or stealing of oil by the workers, which cause losses to the company. Hence, an IoT based palm oil extraction rate monitoring system was developed for heating palm oil in storage tank which considered the effect of temperature on the volume expansion of the tank under heating, density properties of the oil and the content of the oil varying with temperature. The oil storage tank would experience expansion during heating while the varying temperature could cause hydrolysis or solidification of oil. This project used ultrasonic level sensor connected to Arduino UNO for oil level measurement. The upload of raw data to ThingSpeak server via ESP8266 Wifi module, and further data analysis and real-time data display in Microsoft Power BI were successful. The experiment was carried out and the system prototype showed promising result which established the significant impact of the factor of tank expansion and palm oil density properties during heating on the weight of extracted oil and OER. The results further demonstrated the financial loss to the palm oil mill of 14.20% of the total production. Therefore, the system can be applied as a part of palm oil mill automation system to reduce the substantial production loss from a business perspective.

TABLE OF CONTENTS

DECLARATION			**i**
APPROVAL FOR SUBMISSION			**ii**
ACKNOWLEDGEMENTS			**iv**
ABSTRACT			**v**
TABLE OF CONTENTS			**vi**
LIST OF TABLES			**ix**
LIST OF FIGURES			**x**
LIST OF SYMBOLS / ABBREVIATIONS			**xii**
LIST OF APPENDICES			**xiii**
CHAPTER			
1	**INTRODUCTION**		**1**
	1.1	General Introduction	1
	1.2	Importance of the Study	2
	1.3	Problem Statement	2
	1.4	Aim and Objectives	3
	1.5	Scope and Limitation of the Study	4
	1.6	Contribution of the Study	4
	1.7	Outline of the Report	4
2	**LITERATURE REVIEW**		**6**
	2.1	Introduction	6
	2.2	Overall System	6
		2.2.1 Summary	9
	2.3	Liquid Level Sensors	11
		2.3.1 Types of Ultrasonic Sensors	14
	2.4	Temperature Sensor	16
		2.4.1 Summary	18
	2.5	Main Development Board	18
		2.5.1 Summary	20
	2.6	Summary	21

3	**METHODOLOGY AND WORK PLAN**			**22**
	3.1	Introduction		22
	3.2	Materials Selection		22
		3.2.1	Level Sensor	22
		3.2.2	Temperature Sensor	23
		3.2.3	Main Development Board	24
	3.3	Conceptual Design		24
		3.3.1	Hardware Design	24
		3.3.2	Software Design	27
			3.3.2.1 Software Design Calculation	27
	3.4	Project Planning		30
		3.4.1	Flow Chart	30
		3.4.2	Gantt Chart	31
	3.5	Summary		31
4	**RESULTS AND DISCUSSIONS**			**32**
	4.1	Introduction		32
	4.2	Results of Testing the Development Board and External Components		32
		4.2.1	Level Sensing: Parallax PING))) Ultrasonic Level Sensor	33
			4.2.1.1 Liquid Level Measurement before Level Sensor Calibration	33
			4.2.1.2 Liquid Level Measurement after Level Sensor Calibration	34
			4.2.1.3 Level Sensing Results	36
		4.2.2	Temperature Sensing: DS18B20 Sensor	37
			4.2.2.1 Temperature Sensing Result	37
			4.2.2.2 Calculation of Specific Gravity using Temperature Value	37
			4.2.2.3 Calculation of Temperature Factor using Temperature Value Measured	38
		4.2.3	Computation of Oil Weight Stored in Tank	39

	4.3	System-to-Cloud Data Transfer		39
		4.3.1	Connection of Arduino UNO to Internet via ESP8266 Wifi Module	40
		4.3.2	Data Upload to ThingSpeak	40
	4.4	Data Display in Microsoft Power BI		42
		4.4.1	Overview of Power BI Report	42
		4.4.2	Impact of Specific Gravity and Temperature Factor	49
			4.4.2.1 Error in Volume Measurement	49
			4.4.2.2 Error in Extracted Oil Weight, OER and CPO Production Computation in Real-World Application	51
	4.5	Summary		54
5	**CONCLUSIONS AND RECOMMENDATIONS**			**55**
	5.1	Conclusions		55
	5.2	Recommendations for future work		55
REFERENCES				**57**
APPENDICES				**62**

LIST OF TABLES

Table 2.1: Summary of Liquid Monitoring System. 9

Table 2.2: Comparison between Ultrasonic Sensor, Differential Pressure Level Sensor and Radar Sensor. 13

Table 2.3: Comparison between HC-SR04, Parallax PING))) and SRF02 Ultrasonic Sensor. 16

Table 2.4: Comparison between Waterproof DS18B20 Temperature Sensor and Waterproof LM35 Temperature Sensor. 18

Table 2.5: Comparison between Arduino Uno R3 and NodeMCU ESP8266 Development Board. 20

Table 3.1: Preferred Level Sensor (Parallax PING)))) Specification. 22

Table 3.2: Preferred Temperature Sensor (Waterproof DS18B20) Specification. 23

Table 3.3: Preferred Microcontroller Board (Arduino UNO) Specification. 24

Table 4.1: Result Data of Actual Liquid Level and Measured Liquid Level by Level Sensor before Sensor Calibration. 33

Table 4.2: Result Data of Actual Liquid Level and Measured Liquid Level by Level Sensor after Sensor Calibration. 34

Table 4.3: Data Entry of FFB Input Weight, Extracted Oil Weight and OER by Date and Time into Power BI. 46

Table 4.4: Data Entry of OER and Production of Extracted Oil by Date and Time. 47

Table 4.5: Apparent Volume, Actual Volume and Error in Volume Measurement by Time. 50

Table 4.6: Error in Volume Measurement, Gain Weight of Extracted Oil, OER and CPO Production in Real-world Scenario. 53

LIST OF FIGURES

Figure 2.1: Parallax PING))) Sensor (Faisal, 2018). 15

Figure 2.2: HC-SR04 Ultrasonic Sensor (Cytron, 2013). 15

Figure 2.3: SRF02 Ultrasonic Sensor (DFRobot, n.d.). 15

Figure 2.4: Stainless Steel Waterproof DS18B20 Temperature Sensor (Robot-R-Us, n.d.). 17

Figure 2.5: Waterproof LM35 Temperature Sensor (GRobotronics, n.d.). 17

Figure 2.6: Arduino UNO R3 (Faisal, 2018). 19

Figure 2.7: NodeMCU Module (Charles, et al., 2018). 20

Figure 3.1: System Hardware Implementation Diagram. 25

Figure 3.2: Experimental Setup for Liquid Level Measurement.. 26

Figure 3.3: Closer View of Experimental Setup for Liquid Level Measurement. 26

Figure 3.4: System Software Implementation Diagram. 27

Figure 3.5: Flowchart of Project Planning. 30

Figure 3.6: FYP 1 Gantt Chart. 31

Figure 3.7: FYP 2 Gantt Chart. 31

Figure 4.1: Hardware Circuit Connection. 32

Figure 4.2: Graph of Measured Liquid Level by Parallax PING))) Ultrasonic Level Sensor before Sensor Calibration against Actual Liquid Level. 34

Figure 4.3: Graph of Measured Liquid Level by Parallax PING))) Ultrasonic Level Sensor after Sensor Calibration against Actual Liquid Level. 35

Figure 4.4: Level Sensing Result Displayed in Excel Sheet. 36

Figure 4.5: Volume of Stored Oil Displayed in Excel Sheet. 36

Figure 4.6: Temperature Sensing Result Displayed in Excel Sheet. 37

Figure 4.7: Specific Gravity Displayed in Excel Sheet. 38

Figure 4.8: Temperature Factor Displayed in Excel Sheet. 38

Figure 4.9: Weight of Stored Oil Displayed in Excel Sheet. 39

Figure 4.10: Finalised System Software Implementation Diagram. 39

Figure 4.11: Serial Monitor of Test Program for ESP8266 Connection. 40

Figure 4.12: Serial Monitor of Test Program for Sensor Data Uploaded to ThingSpeak. 41

Figure 4.13: Channel View with Uploaded Sensor Data in ThingSpeak. 42

Figure 4.14: Homepage of Power BI Report on OER Monitoring System. 45

Figure 4.15: Line Chart Displaying Volume of Oil Stored by Time. 45

Figure 4.16: Visualization Displaying Relationship between FFB Input Weight and Extracted Oil Weight with OER. 46

Figure 4.17: Visualization Displaying Palm Oil OER and Production by Time. 47

Figure 4.18: Visualization Displaying MPOB Monthly Average OER of CPO for Past Five Years. 48

Figure 4.19: Visualization Displaying Daily CPO Price in Year 2020. 48

Figure 4.20: Visualization Displaying MPOB Monthly Average CPO Prices for Past Five Years. 49

Figure 4.21: Power BI Dashboard showing real-time data streaming of stored oil volume and tank temperature. 49

Figure 4.22: Report Page Showing Error in Volume Measurement. 50

Figure 4.23: Visualization Displaying Error in Volume, OER, Gain Weight of Extracted Oil and CPO Production by Time in Real-world Application. 53

LIST OF SYMBOLS / ABBREVIATIONS

D	distance between sensor and palm oil surface, cm
h	actual level of the palm oil stored in the tank, cm
l	distance between the sensor and the bottom of the oil storage tank, cm
R^2	coefficient of determination
SG_{oil}	specific gravity of palm oil stored in the tank at certain temperature, T_f
t	time taken for the transmitted pulse to be reflected from oil surface and received by receiver, μs
TF	temperature factor of volume expansion of tank
v	speed of sound, cm/μs
V_o	initial volume of the tank before heat expansion, m^3
V_{oil}	volume of palm oil stored in the tank, m^3
W	weight of palm oil stored in the tank
α	homogeneous void fraction
β_{tank}	volumetric temperature expansion coefficient of the tank, $°\text{C}^{-1}$
ΔV_{tank}	change in tank volume due to heating, $V_f - V_o$, m^3
η	pressure ratio
ρ	density, kg/m^3
$\rho_{palm\ oil}$	density of palm oil, kg/m^3 at certain temperature, T_f
ρ_{water}	density of water at 4 °C, kg/m^3
ω	compressible flow parameter
CPO	crude palm oil
CPKO	crude palm kernel oil
FFB	fresh fruit bunches
OER	oil extraction rate
IDE	Arduino Integrated Development Environment

LIST OF APPENDICES

APPENDIX A: Datasheet **62**

DatasheetA-1: Technical Features of HC-SR04 62

DatasheetA-2: Features and Key Specifications of Parallax PING))) Ultrasonic Distance Sensor 63

DatasheetA-3: Introduction and Specifications of SRF02 ultrasonic sensor SKU SEN0005 64

DatasheetA-4: Features, Specifications and Usage of DS18B20 Temperature Sensor 65

DatasheetA-5: Features, Applications and Description of LM35 Precision Centigrade Temperature Sensors 66

DatasheetA-6: Technical Specification of Arduino UNO ATmega328 Microcontroller Development Board 67

DatasheetA-7: Technical Specifications of NodeMCU ESP8266 Microcontroller Development Board 68

APPENDIX B: System Program Code **69**

Program CodeB-1: Program Code in Arduino IDE for Data Upload to Thingspeak 69

Program CodeB-2: Program Script for Power BI Real-time Data Streaming of Volume of Stored Oil and Tank Temperature from ThingSpeak Channel 75

CHAPTER 1

INTRODUCTION

1.1 General Introduction

Having emerged as a major economic crop feeding the world today, oil palm industry is one of the fastest growing sectors in Malaysia, in which Malaysia is currently the second largest palm oil producer in the world (Zulkefli, et al., 2017). In Malaysia, the oil palm industry plays a crucial role in developing the agricultural sector and economy of the country. There are mainly two types of oil produced by oil palm, including crude palm oil (CPO) from the fibrous mesocarp and crude palm kernel oil (CPKO) from the seeds or kernels. CPO is extracted from fresh fruit bunches (FFB) by a grinding process and the percentage is known as oil extraction rate (OER). The national OER represents the overall milling efficiency in which the CPO output is measured against the FFB input into the palm oil mills in Malaysia (Sarip, et al., 2016). According to Malaysian Palm Oil Board (2020), the average OER in eleven states of Malaysia from 2015 until March of 2020 have fluctuated from a high of 22.40 % in 2015 to a low of 18.04 % in 2020. The sliding prices of CPO is pressured by escalating stockpiles and production as well as decelerating exports (Aruna, 2018). In the times of decreasing OER and low prices of crude palm oil (CPO), producers have to improve the performance of OER as it is a management tool in assessment of the mill and plantation performance (Zulkefli, et al., 2017).

The extracted CPO is stored in a storage tank, which is an important part of the palm oil mill as it keeps CPO production save from contaminants which can degrade the oil quality. Nowadays, manual method of monitoring height and temperature monitoring of CPO in storage tank at palm oil mill is still used and done by workers twice a day. The manual monitoring technique is considered to have lower accuracy as the temperature of storage tank will have effect on the tank volume expansion, the constituents of CPO which has three main components (a mixture of water and oil, water in oil emulsions and oil in water emulsions) and the specific gravity of oil in varied temperature (Andrew, n.d.). Hence, this project takes these issues into consideration to simulate the design of OER monitoring system in real-time with an interface to allow user to obtain the collected date from cloud server.

1.2 Importance of the Study

As one of the key parameters which has direct relation to an oil palm enterprise's profitability (Chang, Sani and Basran, 2003), OER can be used to assess the performance of a mill or plantation. Being affected by the amount of oil achieved per hectare of land under cultivation, a higher percentage in the OER indicates that more oil can be extracted from the fruits (Loke, 2013). Decreasing OER will reduce the price for each metric tonne (1,000 kg) of the oil palm fruits produced, shrinking the income of the smallholders whose livelihood depends solely on oil palm output.

The losses of oil in palm oil mill can lead to lower OER, in which the losses could be caused by the theft of CPO. According to Perumal (2009), the increasing theft of CPO forced the industry player to identify the measures to combat the problem. There is lack of study in monitoring system for palm oil extraction rate which take the temperature of CPO storage tank into consideration. The most important process in the storage tank is controlling oil temperature with heating system in order to reduce the water content and the oxygen in the tank (Hafiz, et al., 2016). According to Suda and Keisch (n.d.), changes in temperature will affect the measurement of volume in several ways including the tank dimension and the density of the liquid which vary with temperature. The storage tank itself will experience volume expansion with varying temperature, which will affect the measurement of the exact volume in the tank.

Based on the stated problems, this study is important as it considers the effects of temperature on the storage tank and the palm oil itself in designing the palm oil extraction rate monitoring system. The system is IoT based which allows the user to determine the condition and access the real-time palm oil extraction rate via an interface.

1.3 Problem Statement

In conventional milling process, the storage tank for palm oil is heated with heating system. The temperature of the storage tank can greatly affect the accuracy of the measurement of the oil extraction rate (OER). The water content and the oxygen in the oil, the tank volume expansion under heating as well as the density properties of the oil may vary with temperature. In addition, manual measurement of level of oil in the storage tank is still applied with the use of sounding meters. Sounding meter is a measuring instrument used to determine the height of the oil by placing it from the top of the tank into the bottom of the tank. A calibrated manual book is used to convert the total height of the oil obtained into volume of oil (Hafiz, et al., 2016).

The monitoring of OER using manual method may lead to loss of oil in the tank due to stealing or smuggling of oil by the factory workers, causing the decreasing OER. Moreover, the quality of oil extracted will decrease with the possible contamination between the oil with water, air and impurities in the meter. Manual data monitoring will reduce the accuracy of the data obtained as the result is not real-time, in addition to the waste of unnecessary time and manpower to take the measurement. This monitoring technique is considered as high risk job because workers have to climb a high storage tank of more than 10 m which can threaten their safety.

This project develops a CPO extraction rate monitoring system which can measure real-time data of temperature and the volume of CPO contained in the tank. The results of this measurement is very useful for monitoring the production acquisition, the amount of CPO sold and stored oil. Thus, the company can also do detection if CPO was taken outside the schedule set by the factory.

1.4 Aim and Objectives

The aim of this project is to develop a palm oil extraction rate monitoring system with the storage tank maintained at optimum temperature, taking into consideration the purity of palm oil, tank volume expansion and density properties of oil.

The objectives of the project:

- To maintain the storage tank at optimum temperature to prevent oil hydrolysis and solidification.
- To consider the volume expansion of the storage tank under heating to improve measurement accuracy.

- To determine the density properties of the oil at varied temperature.
- To design IoT based system and to act as an interface for user to obtain real-time data of temperature and the volume of crude palm oil in the storage tank as well as the oil extraction rate and oil storage condition.

1.5 Scope and Limitation of the Study

- It is an IoT based system via integration with cloud server
- The storage tank is maintained at a temperature within the range of room temperature to 80 °C.
- Palm oil as the subject of the study.
- The maximum volume of oil in the storage tank is 10 litres.
- The volume of the temperature sensor in the oil is neglected.
- Effect of temperature on level sensors and temperature sensors are neglected.

1.6 Contribution of the Study

This study focuses on the development of an IoT based palm oil monitoring system to improve the current oil mill conventional practice with less transparency and the manual way of acquiring oil level data comes along with several drawbacks. The study also takes the purity of palm oil, tank volume expansion and density properties of oil into consideration while computing weight of extracted oil, which could bring about significant financial impacts to oil mill operators, improving oil mill efficiency.

1.7 Outline of the Report

Chapter 1 of this report provides a general introduction on palm oil industry in Malaysia with the importance of oil extraction rate as well as the consideration of impact of tank volume expansion and oil density properties during heating. The importance of this study with the problems to be investigated in the study, the aim and objectives and the scope and limitation of the study are discussed.

In Chapter 2, the previous works and researches done regarding the topic of IoT and oil monitoring system are reviewed. The subtopics focus on review and comparison of previous works on overall system, liquid level sensor, temperature sensor and main development board respectively. Chapter 3 highlights the methods adopted in the project after review. The project activities including material selection, hardware and software design as well as project planning are included.

Chapter 4 outlines the result obtained along the system development, comprising of the testing of the development board and external components, system-to-cloud data transfer and data visualization in Microsoft Power BI. Lastly, Chapter 5 concludes the outcomes of the project with some suggestions for future work research direction and improvements.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter will cover the literature review for the overall system of existing liquid volume monitoring system in which the methods and design of the system will be discussed. Besides, the level sensor used for palm oil level measurement in this application will be discussed with review of prior design of liquid volume monitoring or level measurement system. A few types of level sensors will be explained together with the justification and comparison between a few level sensors. In addition, temperature sensors used for temperature regulation and maintenance in the project will be covered as well with comparison between a few types of temperature sensor. Furthermore, main development board connecting the sensors will be discussed with literature review of previous relevant monitoring system. The comparison between a few types of microcontroller-based development board will be covered for justification of the development board selection.

2.2 Overall System

Monitoring system for oil extraction rate includes the measurement of oil level in storage tank which will be then computed into volume of oil extracted from a known weight of fresh fruit bunches. Hafiz, et al. (2016) proposed a system of temperature and volume control at crude palm oil (CPO) storage tank with a user interface to control the real-time production. In the study, the temperature of the tank was kept at 40 °C with the consideration of the heat transfer phenomena which can catalyse the hydrolysis of oil in order to improve the quality of CPO. The system involved the use of an ultrasonic sensor for levelling measurement; two temperature sensors for temperature measurement at top and bottom of the tank; filling phase after receiving CPO volume data and circulation phase for even heating; and configuration and report system including automatic data recording and real-time data display.

Faisal (2018) designed a CPO monitoring system using ultrasonic sensor with the implementation of Neural Network algorithm to anticipate illegal oil taking. The system used an ultrasonic sensor connected to Arduino UNO microcontroller to obtain level data of oil stored in a vertical tank with a pipe and solenoid installed below it. The data received by Arduino will be sent to the web server and be saved in database. The data is ready to be processed and scheduled via Neural Network and be displayed to the user. Another system named InteRface for Monitoring wAter tanks (IRMA) was developed by Gama-Moreno, et al. (2016) with the aim to allow the users to be alert of the water level at anytime for the automation of the water tank level control. The system consisted of instrumentation system of an ultrasonic sensor connected to Arduino Microcontroller Board, application module which store and process the data, a mobile user interface over any mobile device that showed user the levels of water tank using Graphic User Interface, and a desktop interface that send or receive data from mobile interface. The user can be notified of the water tank levels via a SMS sent through a GSM network. The result of the system testing showed that the interface would be automatically updated of the volume of water in the tank in unit litres, through the notification system when the device remained connected to WLAN.

Kauffman (n.d.) designed an oil change monitoring system containing oil measuring compartment. The level or quantity of the oil was determined by the oil measurement compartments with two sets of parallel metal wires by measuring the conductivity of the oil which covered their surfaces. The ratio of the output of vertical wire to the output of horizontal wire was directly related to the level of oil which was then computed into oil quantity after being calibrated in the manufactured compartment. The liquid volume monitoring system designed by Husni, et al. (2016), with the aim to tackle oil theft in vehicles, was based on ultrasonic sensor and Arduino microcontroller. While setting up a small wireless network, Bluetooth module was used as a medium of communication between nodes. After receiving sensor data from two child nodes and location data from GPS/GSM/GPRS Shield module, the central node transmitted the data to a server. Volume of liquid inside the tank can be monitored by user using web-based application at anytime and anywhere. The experimental result showed that the rate of system accuracy under normal circumstances was 99.33 % whereas the rate of system accuracy in tilted circumstances (10°) was 84 %.

Crawford, King and Randolph (1992) designed a monitoring system of fluid in a storage tank with the output in terms of mass. The buoyancy of the load on the displacer suspended in the tank was measured. The resultant force acting on the displacer was proportional to the mass of the fluid stored in the tank. Yahya (2013) did a research to design a multilevel storage tank gauging and monitoring system. In this study, a frequency domain based technique was used by vertically inserting short-circuited coaxial geometry in a tank filled with liquid. The adopted technique had some advantages over other techniques including the low cost instrumentation, competitive gauging accuracy, unlimited number of measurable levels and the level content properties monitoring as extra feature. The result showed that the computation of both liquid levels and permittivity can be done with inaccuracy of less than 0.1.

Case (1980) designed an electronic liquid level monitor and controller which consisted of a digital signal indicating changes in impedance of a passive circuit component. A movable ferrite core within the inductor coil could mechanically track the movement of the liquid level, in which the moving core changed proportionally the impedance of the inductor coil. The liquid levels were determined by comparing the digital signal indicating the impedance of inductor coil over a fixed period of time with an electronically stored table of predetermined liquid levels. Furthermore, there were a few liquid storage level monitoring systems applying differential pressure level measurement. Dulphy-Vigor, Ferenczi and Viard (2002) designed a storage level monitoring device by using a sensor to measure the absolute pressure existing in the tank and the differential pressure between a high point and a low point of the tank. Ferretti, et al. (1988) implemented a system to measure liquid level in a tank which continuously measured the values of differential pressure within the tank. The instantaneous liquid level within the tanks was calculated according to the values of differential pressure obtained, by comparing them with a prior collected average liquid level.

Smith (1984) implemented a specific gravity independent gauging system of liquid filled tank. Two pressure sensing devices were placed in the tank at different levels to sense the differential pressure between the sensing devices and the location of the devices in the tank, which was then used to compute the liquid height independently of specific gravity of the liquid. Suthergreen, Cotton and Zingel (1994) designed a monitoring system for filling of liquid in a tank and the ongoing liquid condition which can determine the quantity, temperature and specific gravity of each oil delivery to the tank. The fluid pressure at upper and lower locations inside the tank was determined using a dual in-line air bell structure and the data was processed to determine the quantities of liquid in the tank and produced corresponding outputs.

In the volume measurement based on detection of liquid level proposed by Suda and Keisch (n.d.), the temperature effects on tank dimension, the liquid level and density of liquid were taken into account. Additionally, the response signal of sensor as well as the response of device for detecting liquid level also varied with temperature. Hence, temperature corrections based on thermal expansion test was carried out by determining the temperature correction factors. In the study conducted by Davies (2016), the temperature's effect on dynamic viscosity, flow rate and density of a few vegetable oils were investigated. The experimental result indicated that when the temperature increased, the dynamic viscosity of all vegetable oil including CPO would decrease, showing a negative non-linearly relationship between temperature and viscosity. The greater the viscosity, the flow rate will be lower. Besides, there was a general trend of decreasing density of vegetable with increasing temperature.

2.2.1 Summary

Table 2.1: Summary of Liquid Monitoring System.

System overview	References
Microcontroller, ultrasonic sensor and temperature sensors were used. CPO in tank was maintained at 40 °C and considered heat transfer effect on oil quality.	Hafiz, et al. (2016)
Arduino UNO and an ultrasonic sensor were used. Data were processed via Neural Network algorithm.	Faisal (2018)
Arduino, ultrasonic sensor, application module, mobile user interface and human-computer interface were used with automatic real-time update of liquid volume.	Gama-Moreno, et al. (2016)

Oil quantity was determined with two sets of parallel metal wires by computing the ratio of the output of vertical wire to the output of horizontal wire.	Kauffman (n.d.)
Arduino, ultrasonic sensor, Bluetooth module and GPS/GSM/GPRS Shield module were used. The results showed accuracy of 99.33 % and 84 % in normal and in tilted circumstances (10°) respectively.	Husni, et al. (2016)
Liquid mass was obtained in proportion to the resultant force acting on a displacer suspended in the tank by measuring the buoyancy of load on displacer.	Crawford, King and Randolph (1992)
Tank monitoring was done based on frequency domain using short-circuited coaxial geometry. The results showed inaccuracy of less than 0.1.	Yahya (2013)
Digital signal was used to indicate impedance changes of passive circuit component. Liquid movement was tracked with a movable ferrite core within inductor coil.	Case (1980)
Differential pressure level measurement was adopted to monitor liquid storage level.	Dulphy-Vigor, Ferenczi and Viard (2002)
Comparison between measured differential pressure values with pre-collected average liquid level was done.	Ferretti, et al. (1988)
Liquid level independently of specific gravity was computed using differential pressure sensing devices.	Smith (1984)
Quantity, temperature and specific gravity of oil was obtained by fluid pressure using a dual in-line air bell structure.	Suthergreen, Cotton and Zingel (1994)
Temperature effects on tank dimension, liquid density and level were considered. Temperature correction was performed based on thermal expansion test.	Suda and Keisch (n.d.)
Temperature effects on dynamic viscosity, flow rate and oil density were considered. The results showed negative non-linear relationship between temperature	Davies (2016)

and viscosity whereas when temperature increased, density of oil decreased.	

In summary, most of the existing fluid level or volume measurement and monitoring systems were using methods including ultrasonic, pressure head, buoyancy and impedance. Few studies considered the effect of temperature on volume expansion of tank, quality of oil (hydrolysis of CPO), specific gravity, density and viscosity of oil.

2.3 Liquid Level Sensors

Liquid level sensors use different techniques and methods to measure the height or position of the fluid surface in a container. There are two types of measurement of level available including continuous and point level measurement. Point level sensors can be used to indicate a single discrete liquid level or pre-set a level condition. The sensors usually function as a high alarm which signal a condition of overflow or as a low alarm condition indicator. Instead of at one point, continuous level sensors can provide liquid level measurement within a range, providing level monitoring of entire system. They can produce analogue outputs with direct correlation to the liquid level in the tank, which is more suitable in our application of oil extraction rate monitoring system. The data from level sensor will be then computed into volume. Continuous level measurement can be done by different sensors such as ultrasonic sensor, hydrostatic (pressure) transmitter level sensors and radar sensor.

An ultrasonic sensor sends out an ultrasonic pulse from a piezoelectric transducer, which will be reflected when the pulse hits the liquid surface and back to the sensor. The level of liquid is then determined by measuring the time between the transmitted pulse and reflected echo. Hafiz, et al. (2016) designed a system for controlling temperature and volume at CPO storage tank which used an ultrasonic sensor HC-SR 04 for level measurement. The result in the research showed that the sensor was capable of reading the liquid level but there was quite a lot of noise generated. In CPO monitoring system designed by Faisal (2018), PING ultrasonic sensor was used for CPO height measurement in the tank. Besides, the InteRfac for Monitoring wAter (IRMA) system designed by

Gama-Moreno, et al. (2016) used an ultrasonic sensor which was mounted on the right edge of the water tank. It detected the proximity between the sensor and the liquid surface. Husni, et al. (2016) developed a liquid volume monitoring system based on ultrasonic sensor SRF 02. They used two ultrasonic sensors to capture the liquid depth to increase the data validity. According to Suda and Keisch (n.d.), the studies about effect of temperature in volume measurement showed that the sound waves were affected by temperature variation through affected sound velocity.

Hydrostatic level sensors determine the liquid level readings by the measurement of the pressure between the bottom of the tank where the sensor was submerged and the surface of the liquid. The greater pressure force exerted on the tank indicates greater depth or amount of liquid present. There are a few hydrostatic measurement devices including displacers, bubblers and differential pressure (DP) level sensor. Crawford, King and Randolph (1992) designed a system for monitoring storage tank by measuring the buoyancy force acting on a cylindrical displacer suspended in a storage tank, which was proportional to the liquid mass in the tank. The displacer extended from the tank bottom to the maximum level reached by the liquid. Moreover, Smith (1984) used a differential pressure sensing device in developing a sensing and gauging method for liquid in a tank. Both Ferretti, et al. (1988) and Dulphy-Vigor, Ferenczi and Viard (2002) adopted differential pressure level sensor in designing the system or device for measurement and monitoring of liquid level in a tank. Both studies collected the data obtained and compared them with the predefined average liquid level threshold to determine the instantaneous liquid level in the tank.

Furthermore, a radar sensor is used for liquid level measurement by measuring the time taken for the transmitted radar signal and its reflected echo from the liquid surface to make a complete return trip. The greater the time taken, the greater the distance between the non-contact sensor and the liquid surface, thus lesser volume of liquid in the tank (Edwards and Otterson, 2014). Radar sensor is suitable for applications of great distance level measurement with its capabilities up to 30 meters. In the water level measurement study conducted by Mai and Barjenbruch (2016), radar gauges were used to measure water level at the German North Sea Coast. Brumbi (2002) applied radar technology in designing storage tank level measurement process. He found out that microwaves by the sensor were propagated at the speed of light at free space, thus the measurement was not affected by the atmosphere above the tank. Motzer (2000) designed the A PULS RADAR gauge for level measurement and process control in which new two wire radar gauges was used for non-contact measurement. Despite of the lower power consumption, the sensor was almost independent of the process conditions such as pressure, temperature and product properties.

To justify the selection of the methods of liquid level measurement, a comparison in terms of some highly accessible and relevant characteristics for three types of popular methods, including ultrasonic sensor, differential pressure level sensor and Radar sensor is done in Table 2.2. Information regarding the sensors are acquired based on the selection guide on liquids and bulk solids continuous level measurement by Endress and Hauser (2017).

Table 2.2: Comparison between Ultrasonic Sensor, Differential Pressure Level Sensor and Radar Sensor.

Level Measurement Methods	**Ultrasonic sensor**	**Differential pressure level sensor**	**Radar sensor**
Process temperature	-40 °C to +150 °C	-70 °C to +400 °C	-196 °C to +450 °C
Measuring range	2 cm to 200 cm	From 10 cm	30 cm to 700 cm
Factors affecting sensor function	▪ Foam ▪ Extreme turbulent or boiling surface ▪ Strong build-up at the sensor	▪ Dynamic pressure fluctuations	▪ Foam ▪ Extreme turbulent or boiling surface ▪ Conductive build-up on antenna connection
Factors affecting sensor accuracy	▪ High vapour pressure ▪ Interfering reflections ▪ Fast temperature change	▪ Changing density ▪ Dynamic pressure	▪ Wall effects ▪ Interfering reflections/ signal strength ▪ Extreme pressure changes
Methods of measurement	▪ Direct measurement	▪ Inferred measurement	▪ Direct measurement

	▪ Non-contact method	(based on hydrostatic head pressure) ▪ Contact method	▪ Non-contact method
Cost	▪ Relatively inexpensive as compared to radar sensor	▪ Inexpensive	▪ Expensive
Mounting	▪ Top mounting	▪ Normally mounted at the base of tank	▪ Top mounting

According to Table 2.2, ultrasonic sensor is the most suitable type of level sensor to be used in this application due to its measuring range as well as its indirect contact method with the liquid, in which the level measurement is done without any physical contact. Besides, it provides continuous level measurement and is easy to install and calibrate, in addition to its cost-effectiveness.

2.3.1 Types of Ultrasonic Sensors

Several previous studies showed the use of ultrasonic sensors of different types in the liquid level monitoring system. Faisal (2018) used the Parallax PING))) ultrasonic sensor in the design of crude palm oil monitoring system in vertical tank. The sensor was mounted at the lid of the tank. The Parallax PING))) sensor, as shown in Figure 2.1, can transmit an ultrasonic burst and provide an output pulse corresponding to the time taken for the echo of burst to return to the sensor. The distance between sensor and liquid can be calculated easily by the width measurement of the echo pulse. The range of precise and non-contact measurement of the sensor is from 2 cm to 300 cm. The connection to microcontroller is easy as only one I/O pin is required. It consists of 3 pins including GND, 5V and SIG (Signal (I/O pin)) (Parallax, 2013).

Figure 2.1: Parallax PING))) Sensor (Faisal, 2018).

Hafiz, et al. (2016) used HC-SR04 ultrasonic sensor in the designed system controlling temperature and volume of CPO storage tank. The result showed that there was a lot of noise generated by the sensor during filling phase of CPO, changing the reading of the liquid volume. According to Costa (2016), HC-SR04 was a common non-contact distance measuring module with a distance range of 2 cm to 400 cm. There were 4 pins including VCC (input power), TRIG (Trigger Input), ECHO (Echo Output) and GND (Ground), as shown in Figure 2.2.

Figure 2.2: HC-SR04 Ultrasonic Sensor (Cytron, 2013).

Husni, et al. (2016) used two SRF02 ultrasonic sensors in the liquid volume monitoring system. According to DFRobot (n.d.), the SRF02 is a single transducer ultrasonic range finder in a small footprint PCB, as shown in Figure 2.3. Since SRF 02 only has single transducer to be used for both transmission and reception, SRF02 has higher minimum range as compared to other dual transducer rangers, which varies from around 17 or 18 cm.

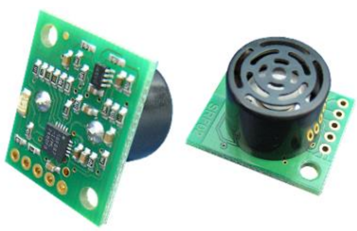


Figure 2.3: SRF02 Ultrasonic Sensor (DFRobot, n.d.).

In summary, Table 2.3 shows the comparison between three different types of ultrasonic sensors including HC-SR04, Parallax PING))) and SRF02 ultrasonic sensor, based on their specifications in which the information are obtained from datasheet of HC-SR04 from the website of Cytron (Cytron, 2013), datasheet of Parallax PING))) sensor from the website of Parallax (Parallax, 2013) and datasheet of SRF02 ultrasonic sensor from the website of DFRobot (DFRobot, n.d.).

Table 2.3: Comparison between HC-SR04, Parallax PING))) and SRF02 Ultrasonic Sensor.

Types of ultrasonic sensors	**Parallax PING)))**	**HC-SR04**	**SRF02**
Operating voltage	+5 VDC	+5 VDC	+5 VDC
Operating current	30 mA typical, 35 mA maximum	15 mA typical, 20 mA maximum	4 mA typical
Operating frequency	40 kHz	40 kHz	40 kHz
Measuring Range	2 cm to 300 cm	2 cm to 400 cm	16 cm to 600 cm
Operating temperature	0 °C to +70 °C	−15 °C to +70 °C	-
Dimension	46 mm x 22 mm x 16 mm	45 mm x 20 mm x 15 mm	24 mm x 20 mm x 17 mm

2.4 Temperature Sensor

A temperature sensor is a temperature measuring device through an electrical signal. The amount of heat energy generated by a system or an object is measured and thus allows the detection of any physical changes to that temperature with the output production in either digital or analogue form. Contact type and non-contact type are two basic physical types of temperature sensors. In this application, temperature sensor is required to maintain the temperature at certain optimum temperature to avoid oil hydrolysis by inserting it into the oil thus a contact type of temperature sensor, which monitor the temperature changes using conduction, will be more suitable. There are a few types of waterproof temperature sensors including DS18B20 and LM35, which will be discussed in this part.

Hafiz, et al. (2016) used both LM35 and DS18B20 for the temperature control system. The result of calibration for temperature sensor using standard temperature and oil bath testing machine showed that LM35 had a lower sensitivity whereas DS18B20 had higher accuracy with more stable temperature readings. Furthermore, Surya and Chauhan (2015) developed a water level indicator which used a LM35 temperature sensor for temperature regulation. The sensor outputs were proportional to the water temperature. The sealed sensor circuitry was not subjected to oxidation and other processes. The result showed that the sensor had low self-heating which would not cause more than 0.1 °C rise in temperature.

The waterproof version of DS18020 digital temperature sensor as shown in Figure 2.4 has unique one-wire interface which allow the communication involving only one port pin. 9 to 12-bit temperature readings is provided over the interface, in which the temperature measuring range is within −55 °C to +125 °C with accuracy of ±0.5 °C. Meanwhile, the LM35 sealed analogue temperature probe as shown in Figure 2.5 can provide precise temperature measurement in wet environments, with the output voltage linearly proportional to the Centigrade temperature. The operating temperature range is from −55 °C to +150 °C.

Figure 2.4: Stainless Steel Waterproof DS18B20 Temperature Sensor (Robot-R-Us, n.d.).

Figure 2.5: Waterproof LM35 Temperature Sensor (GRobotronics, n.d.).

2.4.1 Summary

Table 2.4 shows the comparison between waterproof DS18B20 temperature sensor and waterproof LM35 temperature sensor, based on their features and specifications in which the information of DS18B20 is obtained from the product website (Robot-R-Us, n.d.) whereas the information of LM35 is obtained from the datasheet (Texas Instrument, 2017).

Table 2.4: Comparison between Waterproof DS18B20 Temperature Sensor and Waterproof LM35 Temperature Sensor.

Types of temperature sensors	Waterproof DS18B20	Waterproof LM35
Operating voltage	+3 VDC to +5.5 VDC	+4 VDC to +30 VDC
Temperature range	−55 °C to +125 °C	−55 °C to +150 °C
Communication protocol	One-wire digital	Analogue
Accuracy	±0.5 °C accuracy from −10 °C to +85 °C	1 %
Scale factor	-	0.01V/ °C scale factor
External components	One resistor	None
Stainless steel probe dimension	6 mm Diameter x 30 mm Long	6 mm Diameter x 30 mm Long
Expansion	Multiple sensors can share one pin	Only one sensor for one pin
Cost	Comparatively lower	Comparatively higher

2.5 Main Development Board

Basically, development board is a printed circuit board with circuitry and hardware on-board for facilitation of facilitate experimentation with some microcontrollers. The development board is combined with a processor, memory, chipset and on-board peripherals. The development board has several components including power circuit, programming interface, basic input and output circuitry and input/output (I/O) pins access. There are various types of development boards including Arduino UNO and NodeMCU, which will be discussed in this report.

Arduino UNO R3, as shown in Figure 2.6, is a microcontroller-based development board using ATmega328 microcontroller and it is a prototyping platform with an open source. The microprocessor of the development board can be programmed using C++ language with Arduino Integrated Development Environment (IDE) by connecting it to the computer via an USB cable. Gama-Moreno, et al. (2016) designed monitoring system for a water tank based on mobile devices which used an Arduino UNO R3 microcontroller board to process the data from ultrasonic sensor. According to Husni, et al. (2016), the liquid volume monitoring system involved the use of an Arduino Uno board which was uploaded with specific programs written in the Arduino IDE. Arduino was flexible with a varied inputs and outputs, inexpensive and able to communicate with software running on PC. Besides, Arduino UNO was used in the crude palm oil monitoring system designed by Faisal (2018).

Figure 2.6: Arduino UNO R3 (Faisal, 2018).

As shown in Figure 2.7, the NodeMCU (Node MicroController Unit) is a software and hardware development environment built around ESP8266 System-on-a-Chip (SoC). The ESP8266 module consists of important elements of modern computer including CPU, RAM, networking and a modern operating system and SDK ranging from ESP8266-01 to ESP8266-12 in which ESP8266-12 is the concern for this part (Ashwini, Pavan and Roja, 2018). Mahindar, et al. (2018) used NodeMCU in the IoT-based home appliances control system which included the measurement of water depth from the brim of tank. As the central co-coordinator of the sensors and the actuators, NodeMCU can send and receive data from mobile application via internet server with the built-in support for Wi-Fi connectivity. In the IoT based water level monitoring system designed by

Charles, et al. (2018), the water level readings sensed by the ultrasonic sensor were sent to NodeMCU.

Figure 2.7: NodeMCU Module (Charles, et al., 2018).

2.5.1 Summary

Table 2.5 shows the comparison between Arduino UNO R3 and NodeMCU ESP8266 development board with the information regarding Arduino UNO which is acquired from its datasheet from the product website (RS Components, n.d.) whereas the information regarding NodeMCU is acquired from ESP-12 datasheet from the product website (Joy-it, 2018).

Table 2.5: Comparison between Arduino Uno R3 and NodeMCU ESP8266 Development Board.

Types of development board	Arduino UNO R3	NodeMCU ESP8266
Size	Relatively larger	Smaller
CPU clock frequency	Slower (16 MHz)	Faster (80 MHz to 160 MHz)
CPU Register Width	8-bit	32-bit
Input voltage	7 to 12 V	3 to 3.6 V
Flash Memory Size	Small (32 kB)	Large (4 MB)
RAM	Small (2 kB)	Large (64 kB)
USB port	USB type B connection	Micro USB port
Digital input output ports	14 (6 pulse width modulation (PWM) output)	10

Analogue input pins	6	1
LAN facility	Absent (require external peripherals)	Integrated wireless connection
Development environments	Arduino IDE	Arduino IDE, Lua Loader
Cost	Relatively higher	Low

2.6 Summary

In summary for this chapter, most of the existing liquid level or volume monitoring systems did not take the effect of temperature on liquid quality and container volume expansion into considerations. The methods used for liquid level or volume measurement included ultrasonic sensor, pressure head, coaxial sensor, buoyancy and impedance, which indicated that non-contact ultrasonic based method was the most popular method. Moreover, less system integrated temperature sensor into the design and contact type of temperature sensor was considered and justified. Additionally, the existing systems mostly used Arduino UNO R3 as main development board due to its simplicity, huge community and peripheral supports as well as its cost-effectiveness.

CHAPTER 3

METHODOLOGY AND WORK PLAN

3.1 Introduction

This chapter gives an outline of the project methods referring to journals and research articles. It provides important information and reference that is a criteria for inclusion in the project in order to develop a palm oil extraction rate monitoring system. The planning of design, software, hardware and materials selection are critical steps to build this monitoring system in order to fulfil all the requirements and objectives of the project. Many researches were being done to identify and analyse the problems and data regarding to some relevant systems so this project could be further improved in certain advanced technology development. In addition, an efficient project planning is crucial for development of the system in time and to ensure all aim and objectives are achieved.

3.2 Materials Selection

3.2.1 Level Sensor

In Parallax PING))) ultrasonic sensor, both trigger and response pin are the same, thus it only has three pins: +5V, GND, Trigger. This means that one less pin is used on the Arduino as compared to HC-SR04 sensor. Whereas the minimum measuring range for SRF02 is from 16 cm to 600 cm which is not suitable for the oil level measurement in this project which requires lower minimum measurement range. Therefore, Parallax PING))) ultrasonic sensor is selected for this application.

Table 3.1: Preferred Level Sensor (Parallax PING)))) Specification.

Model	Parallax PING))) (#28015)
Supply voltage	+5 V DC
Supply current	30 mA typical, 35 mA maximum
Operating frequency	40 kHz
Range	2 cm to 3 m

Operating temperature	0 – 70 °C
Input trigger signal	Positive TTL pulse, 2 μs min, 5 μs typical
Output echo signal	Positive TTL pulse, 115 μs min, 18.5 ms max
Dimension	22 mm x 46 mm x 16 mm

3.2.2 Temperature Sensor

Each waterproof DS18B20 temperature sensor can communicate with microcontroller with only one digital pin and multiple sensors can exist on the same one-wire bus thus can share one pin due to unique silicon serial number for each DS18B20. Hence, sensors are allowed to be placed at different location for measurement. Besides, DS18B20 has higher accuracy as compared to LM35. Since DS18B20 temperature sensor is digital, the signal will not be degraded over long distances. Moreover, LM35 sensor is factory calibrated to output voltage which requires calibration in coding while connected to Arduino to output temperature. Therefore, waterproof DS18B20 temperature sensor is chosen for temperature monitoring in this application. However, a resistor is required between voltage and signal pin for DS18B20 sensor.

Table 3.2: Preferred Temperature Sensor (Waterproof DS18B20) Specification.

Model	DS18B20
Operating voltage	+3 VDC to +5.5 VDC
Temperature Range	−55 °C to +125 °C
Accuracy	±0.5 °C accuracy from −10 °C to +85 °C
Resolution	Programmable from 9 to 12-bit
External components	One resistor
Stainless steel tube encapsulation dimension	6 mm (diameter) x 50 mm (length)
Cable diameter	4 mm
Cable Length	1 m
Query time	Less than 750 ms

3.2.3 Main Development Board

NodeMCU ESP8266 is a 3.3 V device thus the usual 5 V sensors would not work without level shifting. In this application, more than one analogue input pins are required thus Arduino UNO is more suitable. Besides, NodeMCU ESP8266 as a 32-bit microcontroller consumes more power than a 8-bit microcontroller (Arduino UNO). However, NodeMCU ESP8266 is integrated with an on-board Wi-Fi feature but this is absent in Arduino UNO. Hence, an external Wi-Fi or Bluetooth module is required to enable wireless transmission. In short, Arduino UNO is chosen as main development board in this application.

Table 3.3: Preferred Microcontroller Board (Arduino UNO) Specification.

Microcontroller	ATmega328
Clock speed (MHz)	16
Program Memory Type	Flash
Flash Memory Size (kB)	32
EEPROM size (kB)	1
SRAM (kB)	2
Operating Voltage (V)	5-12
Programming language	C++
Programming software	Arduino Integrated Development Environment
Digital input output ports	14 (6 PWM output)
Analogue input pins	6
USB port	USB type B connection

3.3 Conceptual Design

3.3.1 Hardware Design

Basically, the hardware is designed as shown in Figure 3.1 below to obtain the level of oil in the storage tank while maintain the tank under certain temperature, and the oil level measurement is to be converted to oil volume using software. Storage tank is cylindrical in shape with heater beneath it for oil heating. At the top of the tank there is a mounting with ultrasonic sensor installed in it. Ultrasonic sensor has a transmitter to transmit pulse signal which is then reflected back by the oil surface, and to be received by the receiver. The time taken for the return of echo is measured by the sensor and the value is returned to the microcontroller as a variable-width pulse. The time is then converted to distance to obtain the oil level.

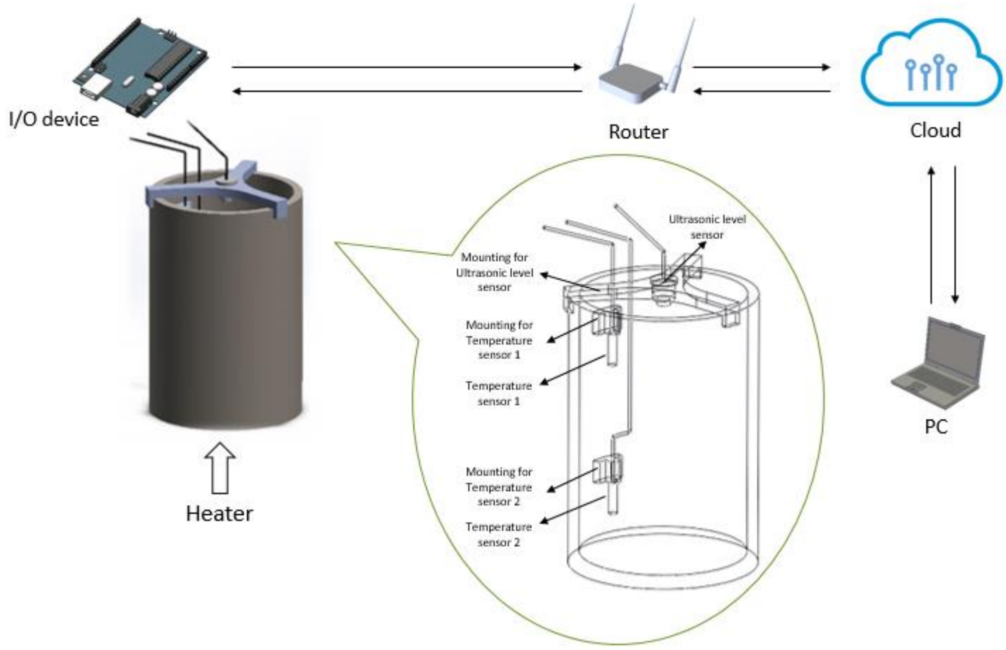


Figure 3.1: System Hardware Implementation Diagram.

The tank is heated and to be maintained under certain temperature for study purpose. There are two sensors mounted against the wall of the tank at bottom and upper part respectively in order to obtain an average temperature for higher accuracy. The sensors are connected to Arduino UNO, which acts as a main controller of the system, and the raw data received will be sent to cloud server to be processed and displayed on application side.

As shown in Figure 3.2 and Figure 3.3, the experiment was carried out to observe the liquid level measurement using Parallax PING))) sensor for both before sensor calibration and after sensor calibration. The ultrasonic sensor was mounted at the middle of the top of the container and it was connected to the Arduino UNO R3 development board. The Arduino UNO was connected to the laptop to be programmed using Arduino IDE. The breadboard was used for circuit connections. A ruler was used as measuring tool for the actual liquid level stored in the tank whereas the experimental liquid level was measured by ultrasonic sensor. The result data for actual liquid level and the corresponding measured liquid level was recorded for both before and after sensor calibration. Then a graph was plotted to determine the strength of the relationship between both measured and actual liquid level by obtaining the coefficient of determination or R^2 value. The experimental result and graphs will be shown in Chapter 4.

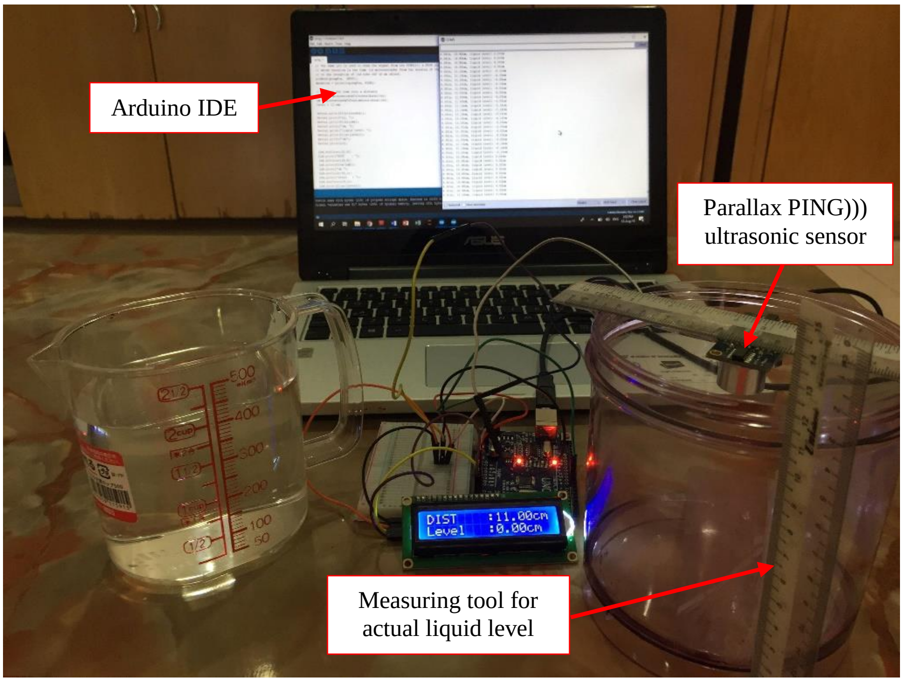


Figure 3.2: Experimental Setup for Liquid Level Measurement..

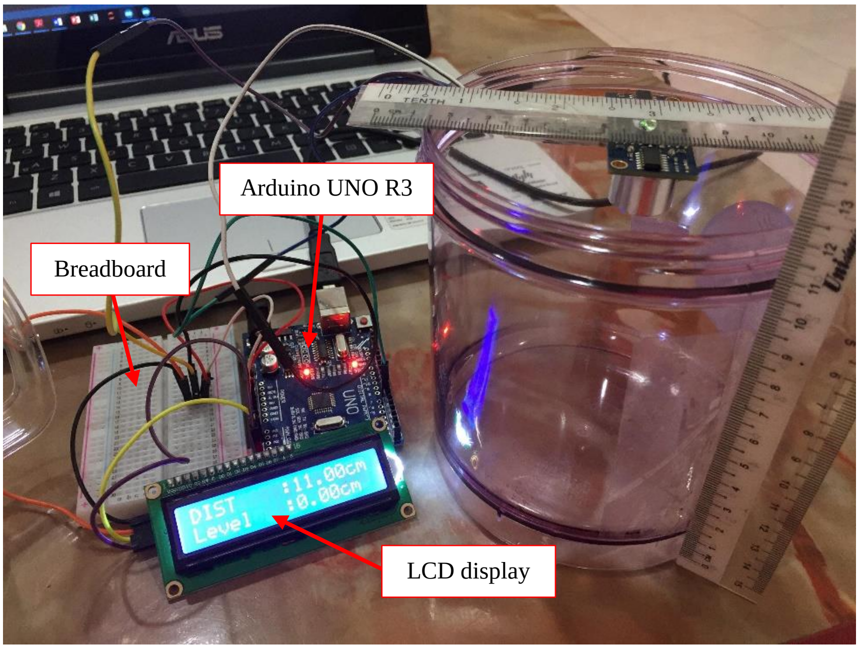


Figure 3.3: Closer View of Experimental Setup for Liquid Level Measurement.

3.3.2 Software Design

Microsoft Azure is used in this project as the cloud computing platform, which includes Azure IoT Hub, Stream Analytics, SQL Storage and Power BI. As demonstrated in Figure 3.4, after the data are acquired from the sensor connected to the Arduino UNO, the raw data are sent to Azure IoT Hub from IoT devices for data digestion. Allowing bi-directional communication between the devices and IoT applications, IoT Hub enables the data to be received from devices and the commands and policies to be sent back to the devices. After that, Azure Stream Analytics as an event-processing engine, is used to examine large amount of data streaming from devices by performing transformations and computations on the data streams. The transformation queries utilize a SQL-like query language for filtering, sorting, aggregating and joining of the streaming data. Then, Stream Analytics routes the job output to SQL Database, which is a cloud storage enabling the users to create different types of storage forms. Lastly, Power BI is used for custom display and visualisation of the data in real-time at user PC.

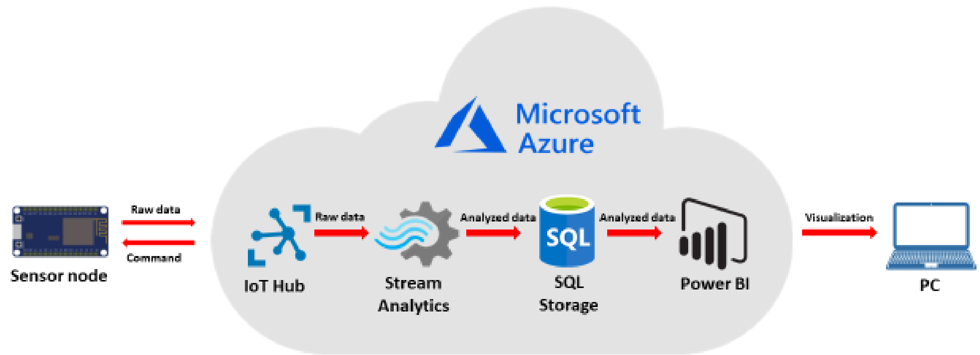


Figure 3.4: System Software Implementation Diagram.

3.3.2.1 Software Design Calculation

The Parallax PING))) ultrasonic sensor is used for palm oil level measurement. It uses sonar to provide high accuracy and stability distance measurement between the sensor and the oil surface. The transmitter will transmit short burst of high pitch sound which will hit the surface of the oil in the tank. The travel speed of the pulse is approximated to the speed of sound in air which is 343m/s. The ultrasound is then reflected from the oil surface whilst the time required for the pulse to go back and fro from the sensor is divided by half. In the Arduino

IDE, time is calculated in microseconds. Hence, the speed of sound is converted to 0.0343 cm/μs. Therefore, the distance is calculated using the formula:

$$D = v \times \frac{1}{2} t \tag{3.1}$$

where
D = distance between sensor and palm oil surface, cm
v = speed of sound, 0.0343 cm/μs
t = time taken for the transmitted pulse to be reflected from oil surface and received by receiver, μs

The actual palm oil level in the tank is then calculated using the formula:

$$h = l - D \tag{3.2}$$

where
h = actual level of the palm oil stored in the tank, cm
l = distance between the sensor and the bottom of the oil storage tank, cm
D = distance between sensor and oil surface, cm

The volume of the palm oil in the cylindrical tank can be then obtained by calculating the product of area of the oil surface and the actual filled level of oil in the tank using the formula (Bird, 2014):

$$V_{oil} = \pi \times r^2 \times 0.01h \tag{3.3}$$

where
V_{oil} = volume of palm oil stored in the tank, m^3
r = inner radius of the tank, m
h = actual level of the palm oil stored in the tank, cm

Moreover, the specific gravity of the palm oil will vary with temperature thus it is taken into consideration. Specific gravity is used to define the density of a fluid as compared to the density of a standard fluid at a specified temperature.

For liquid, water at 4 °C is taken as the standard fluid as this temperature enables water to assume its maximum density. The specific gravity of the palm oil is calculated as (Bansal, 2008):

$$SG_{oil} = \frac{\rho_{palm\ oil}}{\rho_{water}} \tag{3.4}$$

where

SG_{oil} = specific gravity of palm oil stored in the tank at certain temperature, T_f

$\rho_{palm\ oil}$ = density of palm oil, kg/m^3 at certain temperature, T_f

ρ_{water} = density of water at 4 °C which is 1000 kg/m^3

Furthermore, the temperature can have effect on the volume expansion of the storage tank due to heating, which is determined by temperature factor obtained from the formula (Halliday, Resnick and Walker, 2017):

$$\Delta V_{tank} = \beta_{tank} \times V_o \times \Delta T \tag{3.5}$$

$$TF = \frac{\Delta V_{tank}}{V_o} = \beta_{tank} \times \Delta T + 1 \tag{3.6}$$

where

ΔV_{tank} = change in tank volume due to heating, $V_f - V_o$, m^3

β_{tank} = volumetric temperature expansion coefficient of the tank, °C^{-1}

V_o = initial volume of the tank before heat expansion, m^3

ΔT = change in temperature, $T_f - T_o$, °C

TF = temperature factor of volume expansion of tank

In conventional palm oil mill, the production of palm oil is calculated in metric tonnes (MT) as a unit of weight equal to 1000 kilograms (kg). Thus, the volume of the extracted oil, with the consideration of effect of temperature on specific gravity of oil and volume expansion of the storage tank, is calculated using the formula:

$$W(in\ kg) = V_{oil} \times SG_{oil} \times TF \tag{3.7}$$

$$W(in\ MT) = W(in\ kg) \times 0.001 \tag{3.8}$$

where

W = weight of palm oil stored in the tank

V_{oil} = volume of palm oil stored in the tank, m^3

SG_{oil} = specific gravity of palm oil stored in the tank at certain temperature, T_f

TF = temperature factor of volume expansion of tank

Oil extraction rate (OER) is the index to assess the efficiency or profitability of a palm oil mill and as well as the FFB quality (Loke, 2013). OER is simply defined as crude palm oil (CPO) extraction output against the FFB of oil palm input into the mill (Sarip, et al., 2016). The formula for OER calculation is given as:

$$OER = \frac{Gain\ Weight\ of\ Extracted\ Oil}{FFB\ Input\ Weight} \times 100\% \qquad (3.9)$$

3.4 Project Planning

3.4.1 Flow Chart

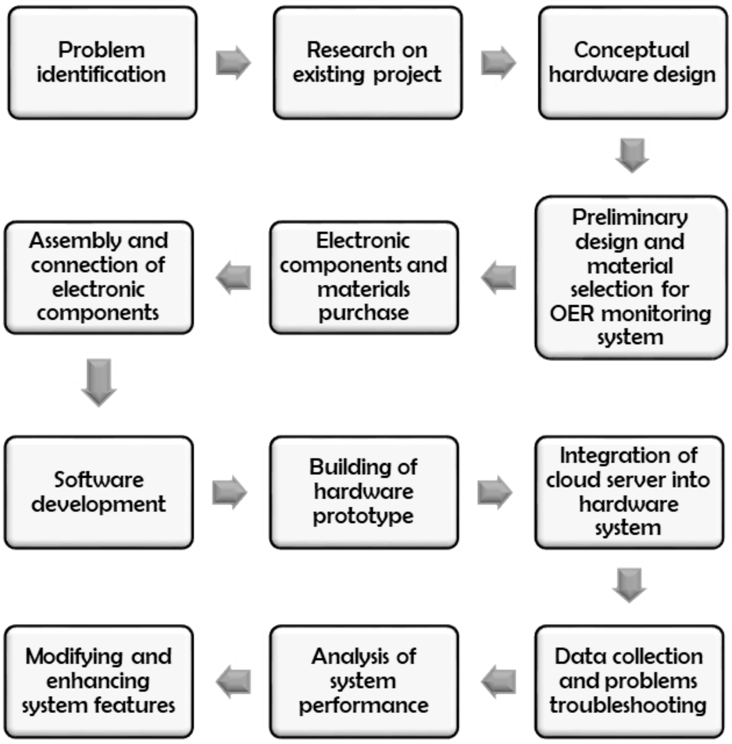


Figure 3.5: Flowchart of Project Planning.

3.4.2 Gantt Chart

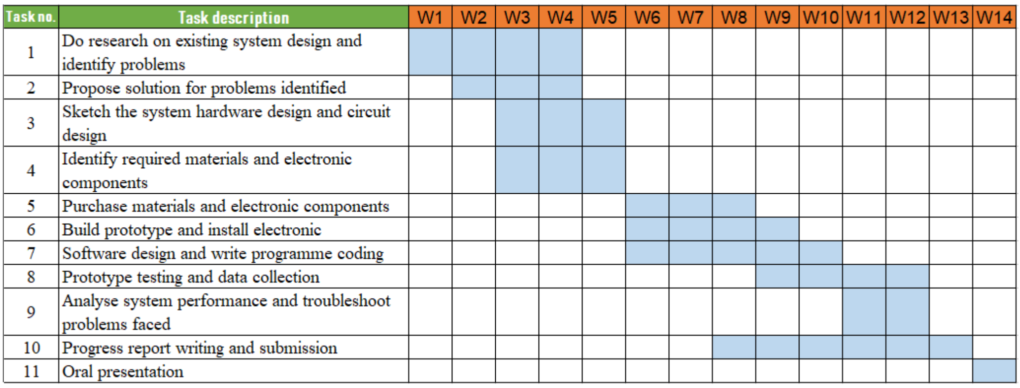


Figure 3.6: FYP 1 Gantt Chart.

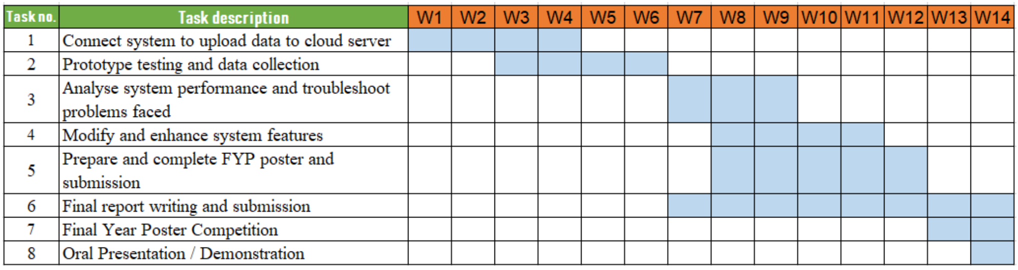


Figure 3.7: FYP 2 Gantt Chart.

3.5 Summary

In short, a well planning is required before a project begins. Planning is very crucial as the priority of work can be determined in order to improve the efficiency of the system design and development. In addition, a project flowchart gives a picture of the steps of the project execution in sequential order which ensures the production of quality work. Gantt chart can be used to track the progress of the project at all time in order to achieve the project milestones at different stages. The system hardware and software design as well as the materials selection are taken into account to produce high quality product which fulfil the aim and objectives of the project.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

This chapter covers the results obtained for testing of Arduino UNO development board and external components including Parallax PING))) ultrasonic level sensor for stored oil level and DS18B20 temperature sensors for tank temperature. An ESP8266 Wifi module was used to enable network connection between development board and IoT platform for sensor data upload. The results were stored in ThingSpeak IoT platform and Microsoft Power BI. Several visualization tools in Power BI were used to provide interactive data visualizations and analytic service.

4.2 Results of Testing the Development Board and External Components

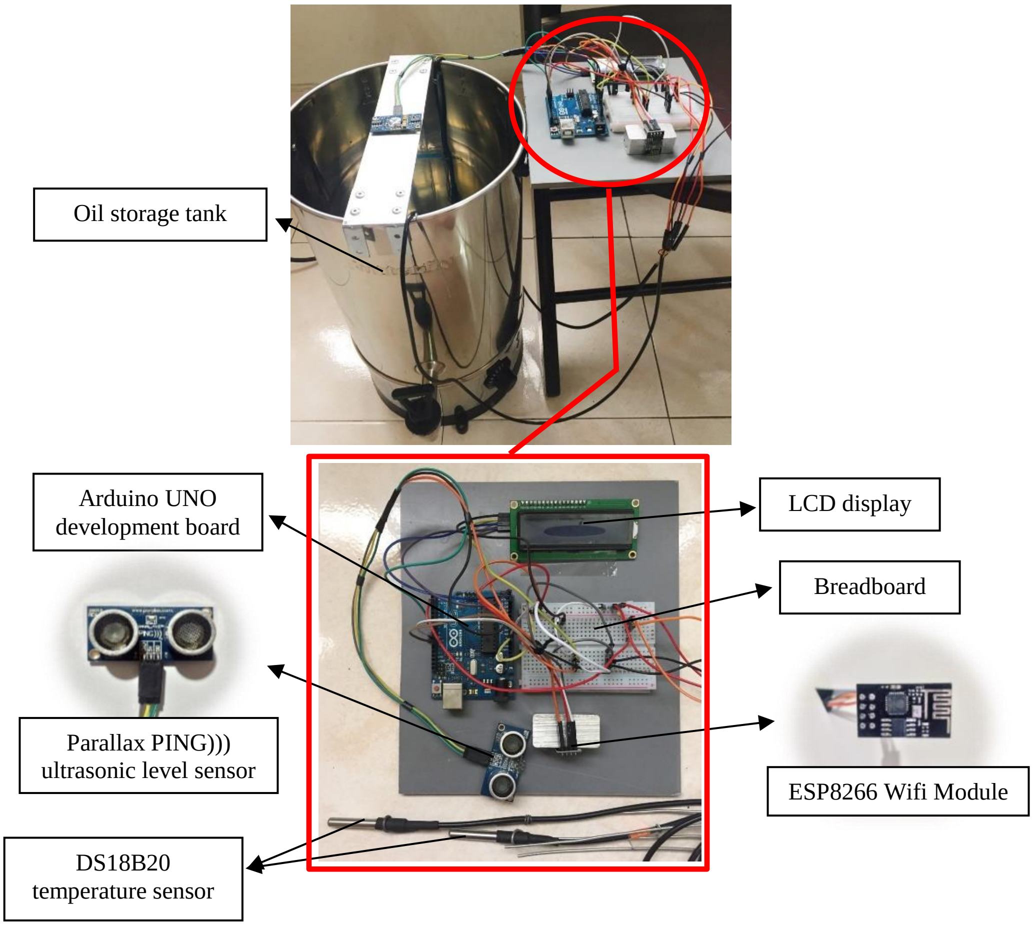


Figure 4.1: Hardware Circuit Connection.

4.2.1 Level Sensing: Parallax PING))) Ultrasonic Level Sensor

4.2.1.1 Liquid Level Measurement before Level Sensor Calibration

The Parallax PING))) ultrasonic level sensor was used to measure the level of water in a container. The result of measured liquid level from sensor and the actual liquid level were collected as shown in Table 4.1 below. The graph of measured liquid level against actual liquid level was plotted in Figure 4.2. This experiment was carried out before any sensor calibration. The linear equation obtained from the graph was y = 0.9389 x + 0.4003 and the coefficient of determination, R^2 value obtained was 0.9987. R^2 value is a statistical measure that describe how close the data are to the fitted regression line. This R^2 value indicated that the model can fit nearly all the data.

Table 4.1: Result Data of Actual Liquid Level and Measured Liquid Level by Level Sensor before Sensor Calibration.

Actual Liquid Level (cm)	Measured Liquid Level by Level Sensor before Sensor Calibration (cm)
0.50	0.84
1.00	1.14
1.50	1.78
2.00	2.29
2.50	2.79
3.00	3.28
3.50	3.76
4.00	4.24
4.50	4.71
5.00	5.17
5.50	5.62
6.00	6.00
6.50	6.48
7.00	6.98
7.50	7.34
8.00	7.83

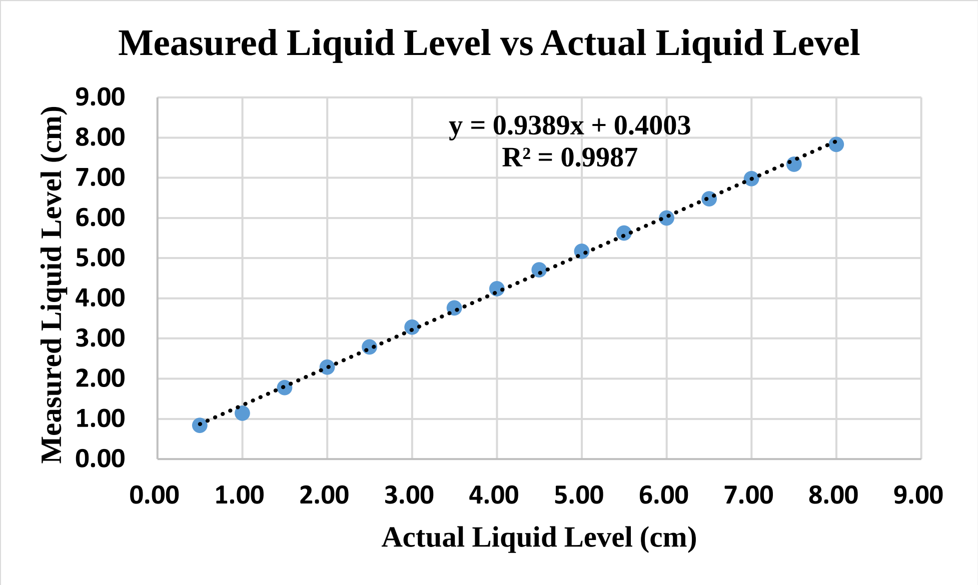


Figure 4.2: Graph of Measured Liquid Level by Parallax PING))) Ultrasonic Level Sensor before Sensor Calibration against Actual Liquid Level.

4.2.1.2 Liquid Level Measurement after Level Sensor Calibration

The result for measured liquid level from Parallax PING))) ultrasonic sensor after sensor calibration and the actual liquid level were collected as shown in Table 4.2 below. The graph of measured liquid level against actual liquid level was plotted in Figure 4.3. The regression line plotted in black dotted line was very close to the points and thus had a high level of fitness. The linear equation obtained from the graph was y = 0.983 x + 0.114 and the coefficient of determination, R^2 value obtained was 0.9999. This R^2 value was higher than that obtained before sensor calibration and this indicated that the model can explain well and predict nearly all future outcomes in a way better than the model before sensor calibration.

Table 4.2: Result Data of Actual Liquid Level and Measured Liquid Level by Level Sensor after Sensor Calibration.

Actual Liquid Level (cm)	Measured Liquid Level by Level Sensor after Sensor Calibration (cm)
0.50	0.60
1.00	1.07
1.50	1.57
2.00	2.10
2.50	2.58
3.00	3.03

3.50	3.57
4.00	4.09
4.50	4.55
5.00	5.02
5.50	5.52
6.00	6.05
6.50	6.53
7.00	6.98
7.50	7.47
8.00	7.94

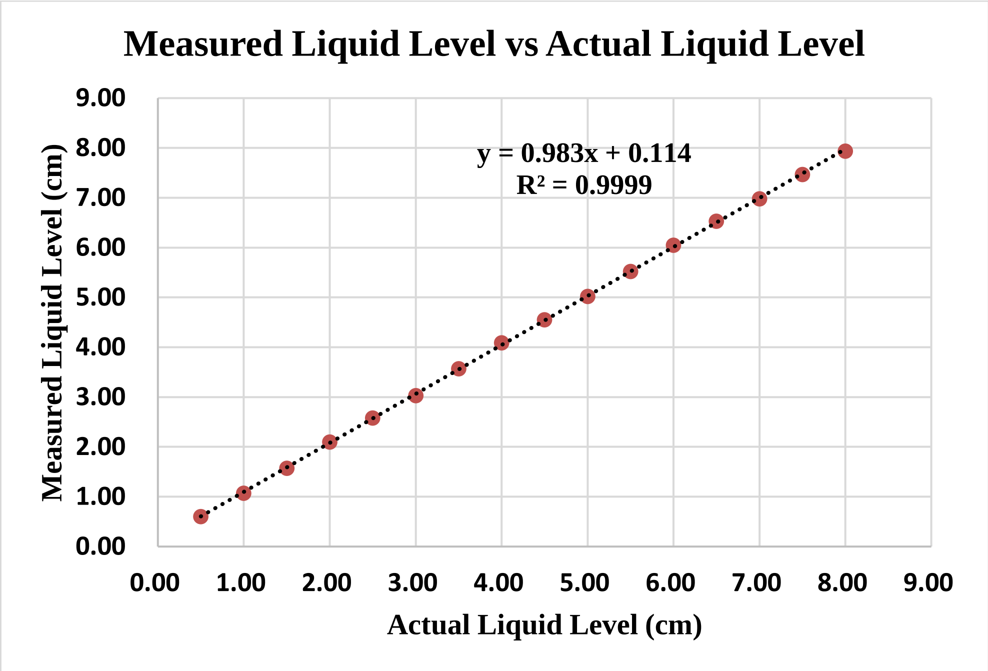


Figure 4.3: Graph of Measured Liquid Level by Parallax PING))) Ultrasonic Level Sensor after Sensor Calibration against Actual Liquid Level.

The result from the experiment showed that the experiment carried out after level sensor calibration had a higher R^2 value, indicating the higher accuracy of the model. The distance between the regression line and the data was very small thus the level of fitness was high. The model is very reliable for future forecasts, which can explained approximately all of the actual liquid level variation.

4.2.1.3 Level Sensing Results

Figure 4.4 shows the level of the oil stored in the tank as the output of level sensor recorded in an excel sheet in which the data were copied from the Serial Monitor of Arduino IDE. The left column of the excel sheet shows the time when the data were taken whereas the most right column shows the oil level in centimetre (cm).

The calibrated level sensor was mounted at the middle of the tank. The distance between the bottom of the tank and the sensor was first measured as 28.8 cm. The value was then deducted by the measured distance between the sensor and oil surface and the calculated value was the oil level stored at different time. The oil level value was applied in the formula as shown in Equation 3.3 to calculate the volume of the stored oil. The result of oil volume was shown in Figure 4.5 below.

	A	B	C
1	Time	Entry ID	Level (cm)
2	2020-03-20 00:49:20 +0800	1	3.32
3	2020-03-20 00:54:27 +0800	2	4.39
4	2020-03-20 00:59:34 +0800	3	5.08
5	2020-03-20 01:04:43 +0800	4	5.78
6	2020-03-20 01:09:47 +0800	5	5.94
7	2020-03-20 01:14:54 +0800	6	6.66
8	2020-03-20 01:20:01 +0800	7	7.63
9	2020-03-20 01:25:08 +0800	8	9.02
10	2020-03-20 01:35:21 +0800	9	13.4
11	2020-03-20 01:40:28 +0800	10	13.51

Figure 4.4: Level Sensing Result Displayed in Excel Sheet.

	A	B	C
1	Time	Entry ID	Volume
2	2020-03-20 00:49:20 +0800	1	1.43881
3	2020-03-20 00:54:27 +0800	2	1.90246
4	2020-03-20 00:59:34 +0800	3	2.20159
5	2020-03-20 01:04:43 +0800	4	2.5082
6	2020-03-20 01:09:47 +0800	5	2.5755
7	2020-03-20 01:14:54 +0800	6	2.88959
8	2020-03-20 01:20:01 +0800	7	3.30837
9	2020-03-20 01:25:08 +0800	8	3.9141
10	2020-03-20 01:35:21 +0800	9	5.81357
11	2020-03-20 01:40:28 +0800	10	5.85844

Figure 4.5: Volume of Stored Oil Displayed in Excel Sheet.

4.2.2 Temperature Sensing: DS18B20 Sensor

4.2.2.1 Temperature Sensing Result

Figure 4.6 shows the excel spreadsheet displaying the average temperature of the oil tank measured by two DS18B20 temperature sensors. One temperature sensor was placed near the bottom of the tank whereas another sensor was placed above the middle height of the tank which was 14.4 cm, acting as threshold level. When the oil level measured was below the threshold level, only the temperature value sensed by the lower sensor was taken as average temperature. Whereas when the oil level measured was above the threshold level, both readings from the sensor were considered to obtain the average temperature. The temperature of the tank was maintained between 40 °C to 65 °C.

	A	B	C
1	Time	Entry ID	Average Temperature
2	2020-03-20 00:49:20 +0800	1	48.75
3	2020-03-20 00:54:27 +0800	2	45.38
4	2020-03-20 00:59:34 +0800	3	51
5	2020-03-20 01:04:43 +0800	4	53.38
6	2020-03-20 01:09:47 +0800	5	51
7	2020-03-20 01:14:54 +0800	6	50.94
8	2020-03-20 01:20:01 +0800	7	61.75
9	2020-03-20 01:25:08 +0800	8	57
10	2020-03-20 01:35:21 +0800	9	45.56
11	2020-03-20 01:40:28 +0800	10	51.38

Figure 4.6: Temperature Sensing Result Displayed in Excel Sheet.

4.2.2.2 Calculation of Specific Gravity using Temperature Value

Specific gravity of palm oil was calculated with the average temperature data input following Equation 3.4. The default specific gravity of palm oil, $\rho_{palm\ oil}$, at different temperature was referred to the list of palm oil properties provided by LIPICO Technologies (2008). The specific gravity obtained was ranged from 0.86862 to 0.87681 and displayed in spreadsheet in Figure 4.7.

	A	B	C
1	Time	Entry ID	Specific Gravity
2	2020-03-20 00:49:20 +0800	1	0.87512
3	2020-03-20 00:54:27 +0800	2	0.87681
4	2020-03-20 00:59:34 +0800	3	0.874
5	2020-03-20 01:04:43 +0800	4	0.87281
6	2020-03-20 01:09:47 +0800	5	0.874
7	2020-03-20 01:14:54 +0800	6	0.87403
8	2020-03-20 01:20:01 +0800	7	0.86862
9	2020-03-20 01:25:08 +0800	8	0.871
10	2020-03-20 01:35:21 +0800	9	0.87672
11	2020-03-20 01:40:28 +0800	10	0.87381

Figure 4.7: Specific Gravity Displayed in Excel Sheet.

4.2.2.3 Calculation of Temperature Factor using Temperature Value Measured

The result of temperature factor of the tank calculated using Equation 3.6 is displayed in excel spreadsheet as shown in Figure 4.8. As the storage tank is made of stainless steel as specified in the product details, the coefficient of volumetric expansion of stainless steel as mentioned by Hughes and Keysar (1963) was 0.000028 °F. While 1 °F = 1.8 °C, the volumetric temperature expansion coefficient of the tank, β_{tank}, was taken as 0.00005 $°C^{-1}$. Meanwhile, room temperature, 25 °C, was taken as reference value for the initial temperature, T_o, of the tank. The temperature factor obtained ranged from 1.00102 to 1.00184.

	A	B	C
1	Time	Entry ID	Temperature Factor
2	2020-03-20 00:49:20 +0800	1	1.00119
3	2020-03-20 00:54:27 +0800	2	1.00102
4	2020-03-20 00:59:34 +0800	3	1.0013
5	2020-03-20 01:04:43 +0800	4	1.00142
6	2020-03-20 01:09:47 +0800	5	1.0013
7	2020-03-20 01:14:54 +0800	6	1.0013
8	2020-03-20 01:20:01 +0800	7	1.00184
9	2020-03-20 01:25:08 +0800	8	1.0016
10	2020-03-20 01:35:21 +0800	9	1.00103
11	2020-03-20 01:40:28 +0800	10	1.00132

Figure 4.8: Temperature Factor Displayed in Excel Sheet.

4.2.3 Computation of Oil Weight Stored in Tank

The weight of stored oil in the tank was calculated based on Equation 3.7 and 3.8 in unit of metric ton (MT), taking the volume of oil with the factor of temperature, which was represented by specific gravity and temperature factor, into consideration. The results were shown in Figure 4.9.

	A	B	C
1	Time	Entry ID	Weight of Extracted Oil (MT)
2	2020-03-20 00:49:20 +0800	1	0.00126
3	2020-03-20 00:54:27 +0800	2	0.00167
4	2020-03-20 00:59:34 +0800	3	0.00193
5	2020-03-20 01:04:43 +0800	4	0.00219
6	2020-03-20 01:09:47 +0800	5	0.00225
7	2020-03-20 01:14:54 +0800	6	0.00253
8	2020-03-20 01:20:01 +0800	7	0.00288
9	2020-03-20 01:25:08 +0800	8	0.00341
10	2020-03-20 01:35:21 +0800	9	0.0051
11	2020-03-20 01:40:28 +0800	10	0.00513

Figure 4.9: Weight of Stored Oil Displayed in Excel Sheet.

4.3 System-to-Cloud Data Transfer

The initial proposed design of software architecture was modified by replacing Microsoft Azure Cloud services including IoT Hub, Stream Analytics and SQL Database with ThingSpeak. ThingSpeak is an open-source platform allowing user to collect and store sensor data to the cloud where separate channels can be created to store the data. ThingSpeak provides free hosting for channel in addition to simple and ready visualizations for stored data. The data was then uploaded to Microsoft Power Bi for further analysis and visualization. The final software architecture of the system was as shown in Figure 4.10.

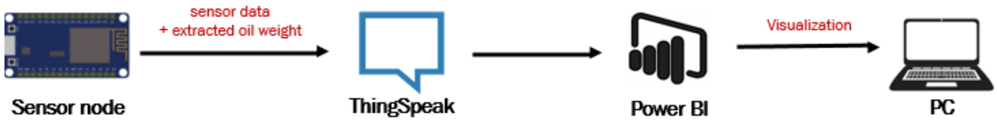


Figure 4.10: Finalised System Software Implementation Diagram.

4.3.1 Connection of Arduino UNO to Internet via ESP8266 Wifi Module

As Arduino UNO development board is not integrated with any Wifi module for network connection, an ESP8266 Wifi module is required to enable internet connectivity to embedded applications. ESP8266 Wifi module is a low cost standalone wireless transceiver to be used for end-point IoT developments.

ESP8266 was programmed using AT commands to enable serial communication between the Serial Monitor and ESP8266 through Arduino UNO. When a command was entered into the serial monitor on the computer, Arduino relayed it to the ESP8266 and when the commands were received, it replied with an acknowledgment. After checking for the connectivity using AT commands, ESP8266 was connected to the ThingSpeak API using TCP Protocol. Figure 4.11 below showed the output of test program for ESP8266 connection to ThingSpeak displayed in Serial Monitor.

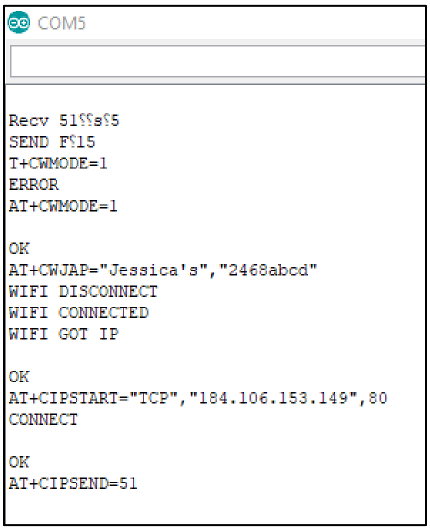


Figure 4.11: Serial Monitor of Test Program for ESP8266 Connection.

4.3.2 Data Upload to ThingSpeak

As the TCP connection between the ESP8266 and ThingSpeak was successfully enabled, the program for uploading Parallax PING))) ultrasonic level sensor data to ThingSpeak was tested. Figure 4.12 showed the Serial Monitor of test program for level sensor data uploaded to ThingSpeak.

The outlook of ThingSpeak channel with raw data uploaded from Arduino UNO via ESP8266 Wifi module was shown in Figure 4.13. There were six fields created to contain six groups of data including the oil level, average temperature, volume of stored oil, specific gravity, temperature factor and weight of stored oil, with simple chart visualization of each field parameter against uploaded time.


```
00:13:57.207 -> AT+RST
00:13:57.207 ->
00:13:57.207 -> OK
00:13:57.207 -> bBɛ̃Scj̃ZRNZRNHNd#Aḩ SD
00:13:57.241 -> □□t%9G□iC□□
00:13:57.274 -> ready
00:13:57.274 -> WIFI CONNECTED
00:13:57.308 -> WIFI GOT IP
00:13:58.282 -> AT+CWMODE=1
00:13:58.282 ->
00:13:58.282 -> OK
00:14:00.314 -> AT+CWJAP="D_Link","abc0164617616"
00:14:00.347 -> WIFI DISCONNECT
00:14:03.328 -> AT+CIPSTART="TCP","184.106.153.149",80
00:14:06.375 -> AT+CIPSEND=54
00:14:06.408 -> GET /update?api_key=VG99NN6D37BRJCCC&field1=162.38cm
00:14:07.480 -> AT+CIPSTART="TCP","184.106.153.149",80
00:14:10.535 -> AT+CIPSEND=54
00:14:10.569 -> GET /update?api_key=VG99NN6D37BRJCCC&field1=183.33cm
00:14:11.640 -> AT+CIPSTART="TCP","184.106.153.149",80
00:14:14.652 -> AT+CIPSEND=54
00:14:14.719 -> GET /update?api_key=VG99NN6D37BRJCCC&field1=183.43cm
00:14:15.792 -> AT+CIPSTART="TCP","184.106.153.149",80
00:14:18.820 -> AT+CIPSEND=54
00:14:18.890 -> GET /update?api_key=VG99NN6D37BRJCCC&field1=183.31cm
00:14:19.938 -> AT+CIPSTART="TCP","184.106.153.149",80
00:14:22.979 -> AT+CIPSEND=54
00:14:23.047 -> GET /update?api_key=VG99NN6D37BRJCCC&field1=183.31cm
00:14:24.092 -> AT+CIPSTART="TCP","184.106.153.149",80
00:14:27.149 -> AT+CIPSEND=54
00:14:27.182 -> GET /update?api_key=VG99NN6D37BRJCCC&field1=183.43cm
00:14:28.253 -> AT+CIPSTART="TCP","184.106.153.149",80
00:14:31.286 -> AT+CIPSEND=54
00:14:31.353 -> GET /update?api_key=VG99NN6D37BRJCCC&field1=183.33cm
00:14:32.430 -> AT+CIPSTART="TCP","184.106.153.149",80
00:14:35.446 -> AT+CIPSEND=54
00:14:35.513 -> GET /update?api_key=VG99NN6D37BRJCCC&field1=183.33cm
00:14:36.552 -> AT+CIPSTART="TCP","184.106.153.149",80
00:14:39.602 -> AT+CIPSEND=54
00:14:39.670 -> GET /update?api_key=VG99NN6D37BRJCCC&field1=182.97cm
00:14:40.707 -> AT+CIPSTART="TCP","184.106.153.149",80
00:14:43.732 -> AT+CIPSEND=54
00:14:43.799 -> GET /update?api_key=VG99NN6D37BRJCCC&field1=183.31cm
```

Figure 4.12: Serial Monitor of Test Program for Sensor Data Uploaded to ThingSpeak.

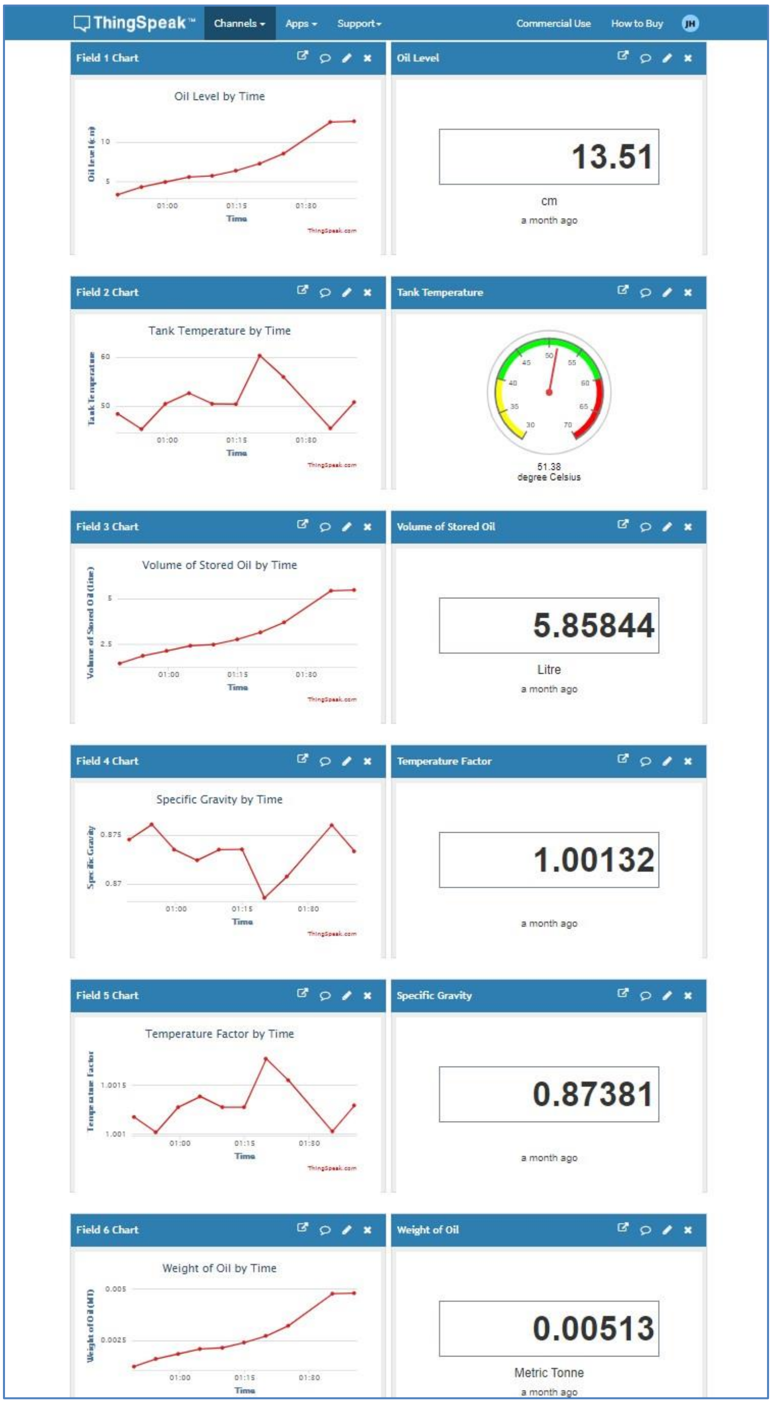


Figure 4.13: Channel View with Uploaded Sensor Data in ThingSpeak.

4.4 Data Display in Microsoft Power BI

4.4.1 Overview of Power BI Report

The raw data from ThingSpeak were uploaded to Power BI for data analysis and visualization. The Power BI report of the overall system consisted of visualizations demonstrating several important parameters, delivering handy and useful information to user. The outlook of the homepage of report in Power

BI for the system was shown in Figure 4.14 below. The top left of the page displaying the title of the system while the card at the top right corner showing the latest CPO price updated daily as acquired from the website of Malaysian Palm Oil Board (MPOB) (Malaysian Palm Oil Board, 2020). Two cards positioned in the middle of the page showed the latest real-time volume of stored oil and the average temperature of the tank.

The simple line chart labelled with “1”at upper left of the page displayed the updated oil volume in the storage tank, as shown in Figure 4.15. The line and clustered column chart with label “2” at the bottom left corner of the report homepage demonstrated the relationship between FFB input weight and gain weight of extracted oil with OER, as shown in Figure 4.16. The higher percentage of oil extracted from the given FFB input resulted in higher value of OER, indicating better quality of the batch of oil palm. Table 4.3 showed the entry data of date, time, FFB input weight, gain weight of extracted oil from respective FFB input batch and OER in Power BI. The dataset of FFB input weight was the manual input data uploaded from excel spreadsheet in CSV format into Power BI. In current palm oil milling process, the input weight data of particular batch of FFB was obtained upon the arrival where the weighbridges are used to weigh the bunches in trucks. According to portal of Malaysian Palm Oil Board (Malaysian Palm Oil Board, 2020) which provided the data record of OER in eleven states of Malaysia from 2016 to March of 2020, OER for crude palm oil fluctuated from a high of 22.40 % in 2015 to a low of 18.04 % in 2020. Hence, the manual input data for FFB input weight was inserted with respect to the OER range by performing back calculation using gain weight of extracted oil. Meanwhile, the gain weight of extracted oil for respective FFB input batch was calculated by retrieving the weight of extracted oil newly added into the tank for each arrival of new batch of FFB input. The computation of OER was completed in Power BI.

The visualization labelled with “3” at the bottom right corner of the homepage in Figure 4.14 was a line and single column chart displaying the OER and production from extracted oil by time, as shown in Figure 4.17 in an enlarged view. The production from oil extracted by respective FFB input was computed by the product of gain weight of extracted oil and the CPO price on that particular date. Table 4.4 displayed the date, time, OER and production with its total represented in Malaysia Ringgit (RM) of entry data. In this study, the CPO price used to compute the production was RM 2311.50 as updated on 23 March 2020, as displayed in the card positioned at the top right corner in Figure 4.14. In addition, the past years records and trends of monthly average OER of CPO in Malaysia were obtained from the website of MPOB (Malaysian Palm Oil Board, 2020), and to be displayed in visualizations as shown in Figure 4.18. This enables the user to observe the trend of the average OER from the stacked area chart as well as acquire preferred dataset from the table. For example, as observed from the chart, the monthly average OER was noticeably higher from August to October for past five years from year 2016 to 2020. These useful information are beneficial for oil palm plantation owners to conduct further study on the reasons behind (such as conducive weather for plantation) for improvement of the oil palm FFB quality.

The multi-line chart at the top right corner of the report homepage with a label “4” illustrated the daily CPO price in the year 2020, as shown in Figure 4.19 for a larger view. Furthermore, the monthly average CPO prices for past five years from year 2016 to 2020 were visualized in the stacked area chart and table as shown in Figure 4.20, with the data obtained from MPOB website (Malaysian Palm Oil Board, 2020).

The Power BI dashboard was created to show the real-time data streaming of volume of stored oil in tank as well as the tank temperature, as displayed in Figure 4.21. Power BI REST APIs method was used to streaming and push the datasets and a Python script was written to read the volume and temperature data from ThingSpeak channel feed and then send to Power BI. This custom data streaming feature enables the user to gain a quick look at the storage condition in the oil tank with the real-time update of oil volume. Hence, oil mill operator is managed to curb crude palm oil thefts in the mill effectively through the real-time monitoring system, by identifying the suspicious or abnormal trend of oil volume. Meanwhile, the user can make sure the maintenance of tank temperature at optimum temperature with the visual updating temperature reading.

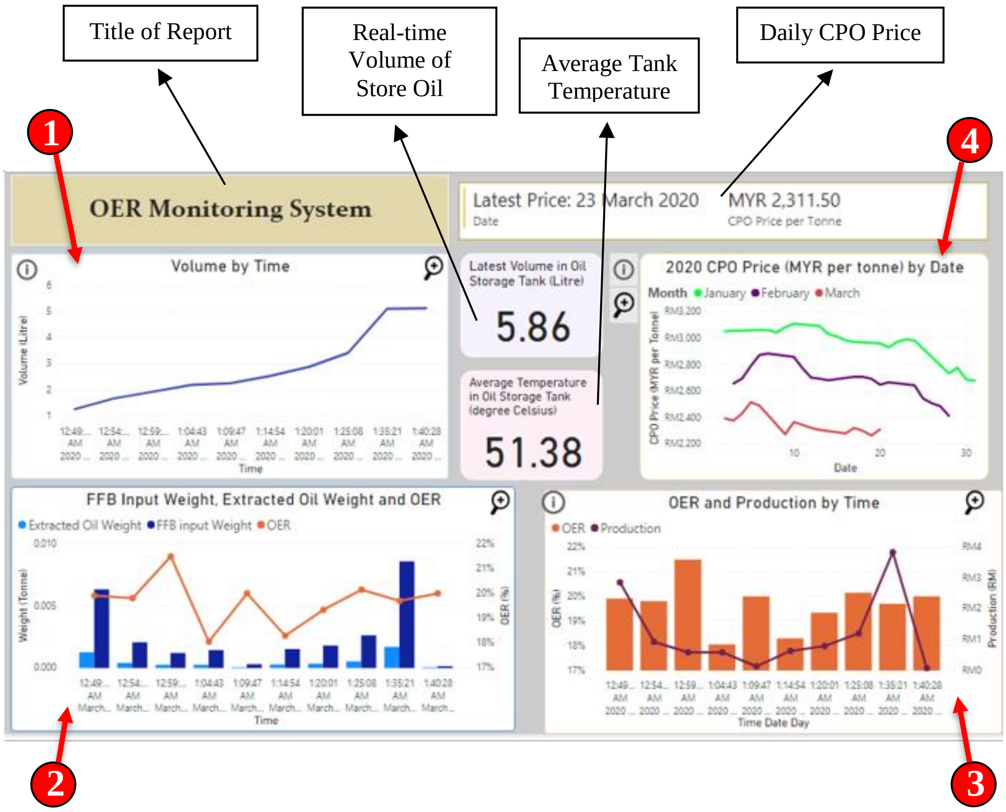


Figure 4.14: Homepage of Power BI Report on OER Monitoring System.

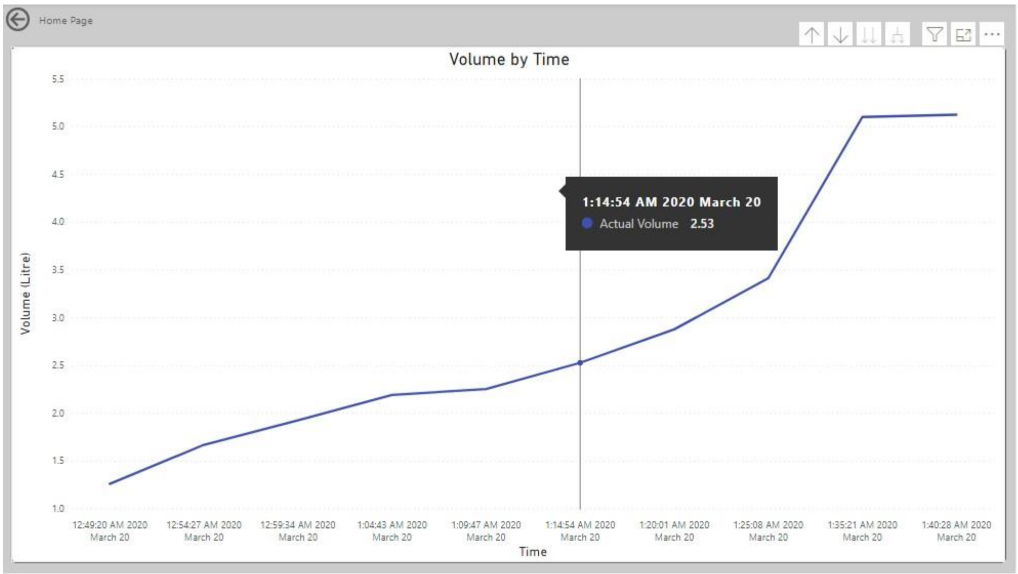


Figure 4.15: Line Chart Displaying Volume of Oil Stored by Time.

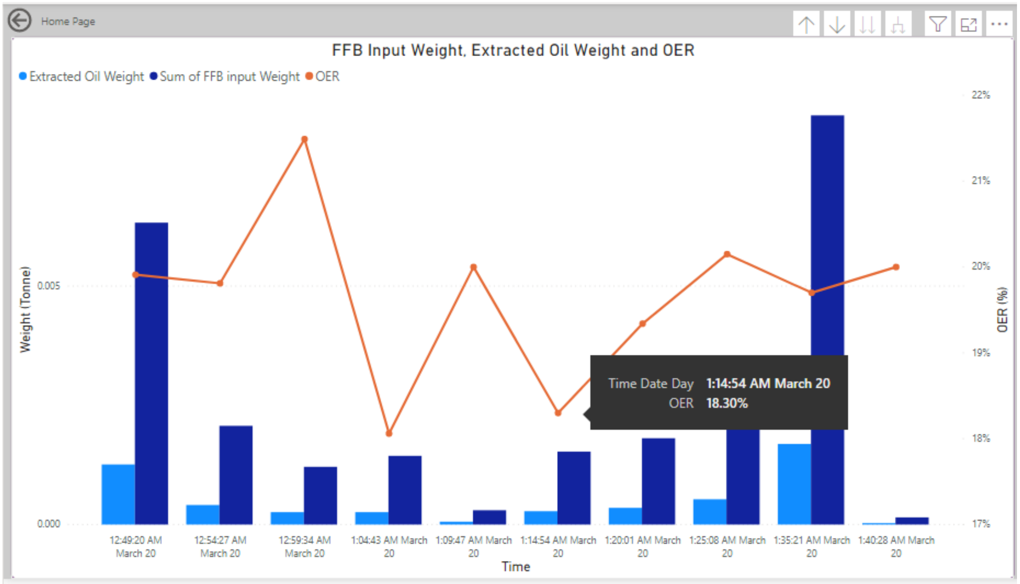


Figure 4.16: Visualization Displaying Relationship between FFB Input Weight and Extracted Oil Weight with OER.

Table 4.3: Data Entry of FFB Input Weight, Extracted Oil Weight and OER by Date and Time into Power BI.

Date	**Time**	**FFB Input Weight (MT)**	**Gain Weight of Extracted Oil (MT)**	**OER**
Friday, March 20, 2020	12:49:20 AM	0.00633	0.00126	19.91%
Friday, March 20, 2020	12:54:27 AM	0.00207	0.00041	19.81%
Friday, March 20, 2020	12:59:34 AM	0.00121	0.00026	21.49%
Friday, March 20, 2020	1:04:43 AM	0.00144	0.00026	18.06%
Friday, March 20, 2020	1:09:47 AM	0.00030	0.00006	20.00%
Friday, March 20, 2020	1:14:54 AM	0.00153	0.00028	18.30%
Friday, March 20, 2020	1:20:01 AM	0.00181	0.00035	19.34%
Friday, March 20, 2020	1:25:08 AM	0.00263	0.00053	20.15%
Friday, March 20, 2020	1:35:21 AM	0.00858	0.00169	19.70%
Friday, March 20, 2020	1:40:28 AM	0.00015	0.00003	20.00%

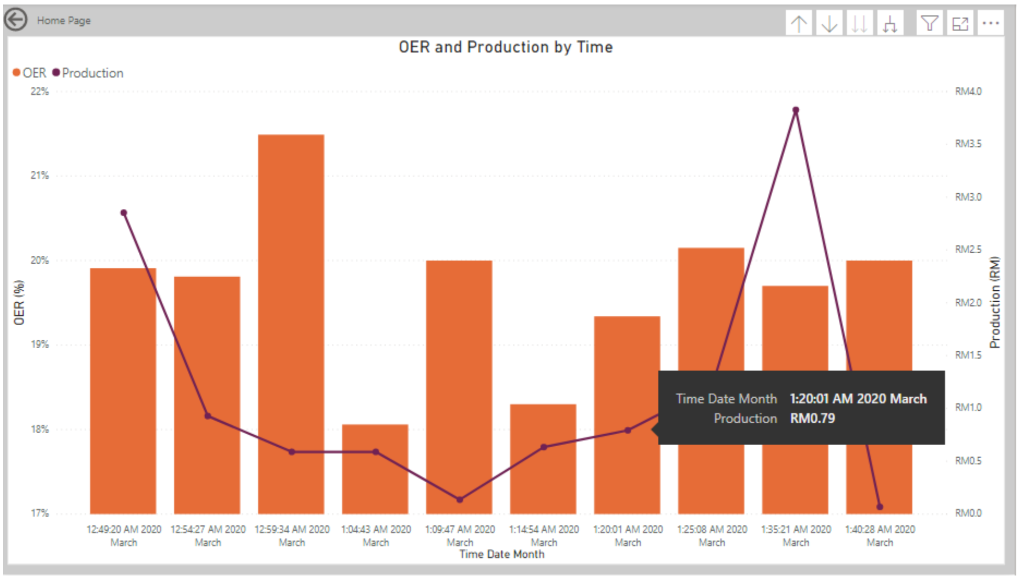
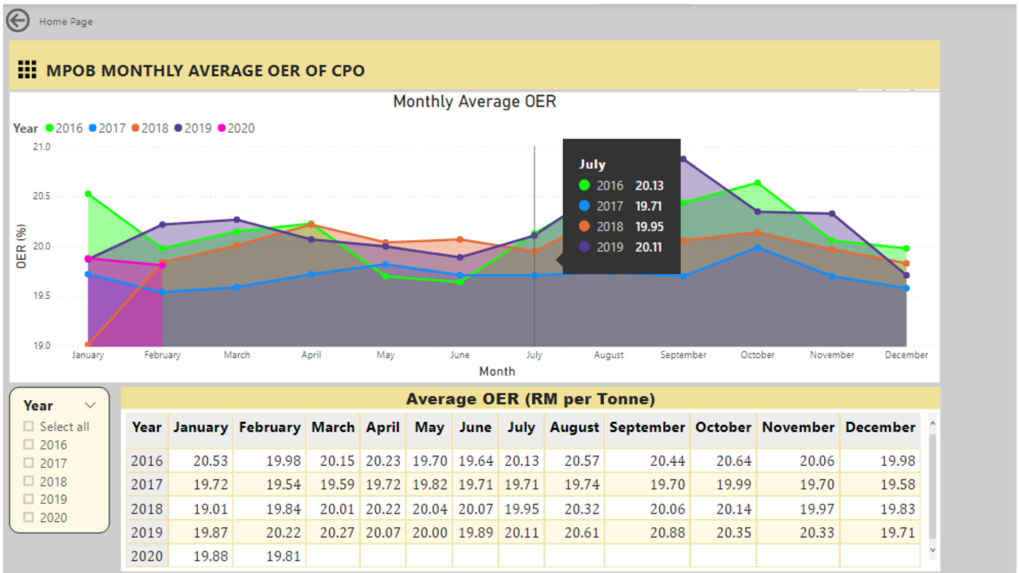


Figure 4.17: Visualization Displaying Palm Oil OER and Production by Time.

Table 4.4: Data Entry of OER and Production of Extracted Oil by Date and Time.

Date	Time	OER	Production (RM)
Friday, March 20, 2020	12:49:20 AM	19.91%	2.85
Friday, March 20, 2020	12:54:27 AM	19.81%	0.93
Friday, March 20, 2020	12:59:34 AM	21.49%	0.59
Friday, March 20, 2020	1:04:43 AM	18.06%	0.59
Friday, March 20, 2020	1:09:47 AM	20.00%	0.14
Friday, March 20, 2020	1:14:54 AM	18.30%	0.63
Friday, March 20, 2020	1:20:01 AM	19.34%	0.79
Friday, March 20, 2020	1:25:08 AM	20.15%	1.20
Friday, March 20, 2020	1:35:21 AM	19.70%	3.83
Friday, March 20, 2020	1:40:28 AM	20.00%	0.07
Total		-	**11.62**



Year	January	February	March	April	May	June	July	August	September	October	November	December
2016	20.53	19.98	20.15	20.23	19.70	19.64	20.13	20.57	20.44	20.64	20.06	19.98
2017	19.72	19.54	19.59	19.72	19.82	19.71	19.71	19.74	19.70	19.99	19.70	19.58
2018	19.01	19.84	20.01	20.22	20.04	20.07	19.95	20.32	20.06	20.14	19.97	19.83
2019	19.87	20.22	20.27	20.07	20.00	19.89	20.11	20.61	20.88	20.35	20.33	19.71
2020	19.88	19.81										

Figure 4.18: Visualization Displaying MPOB Monthly Average OER of CPO for Past Five Years.

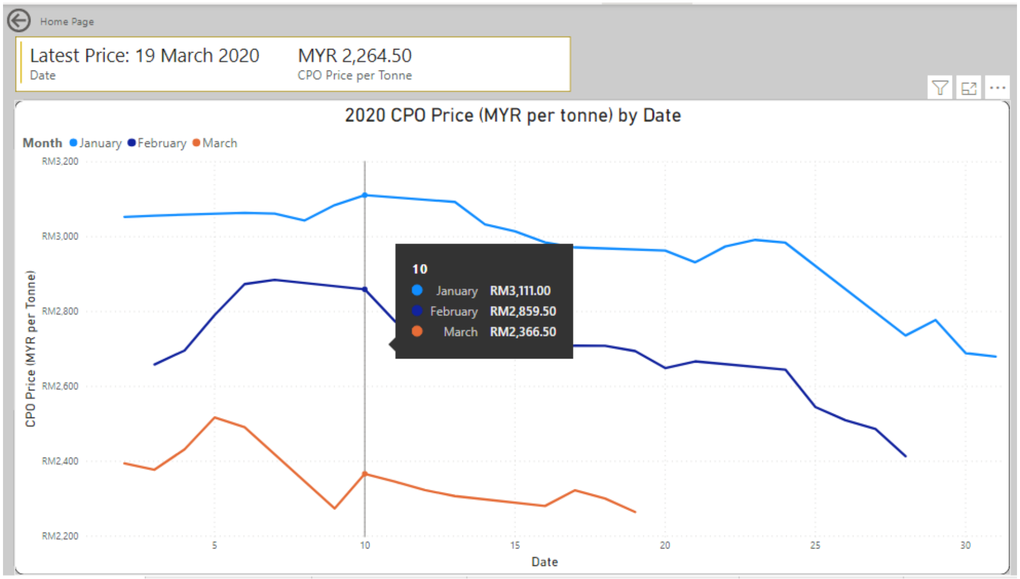
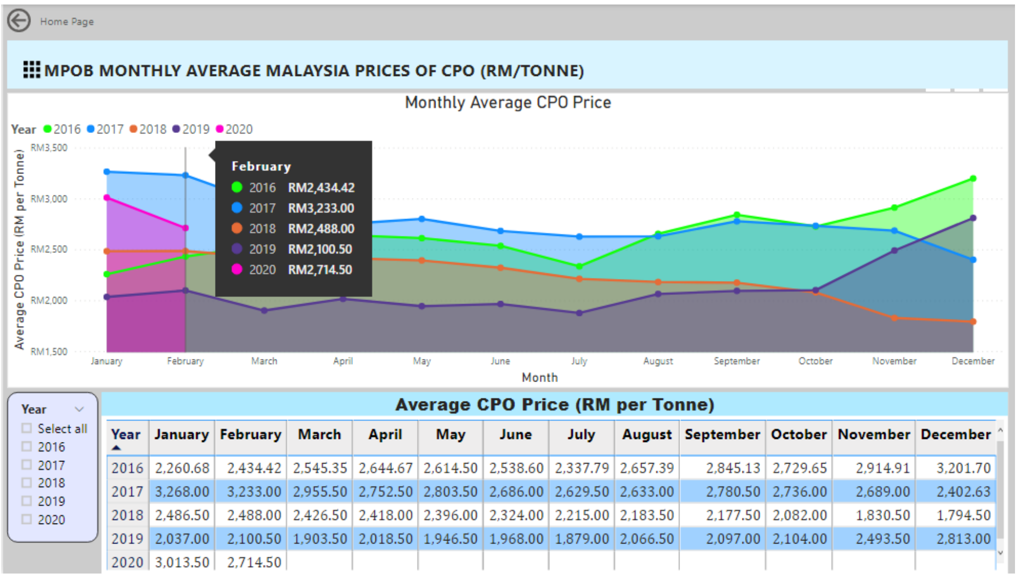


Figure 4.19: Visualization Displaying Daily CPO Price in Year 2020.



Year	January	February	March	April	May	June	July	August	September	October	November	December
2016	2,260.68	2,434.42	2,545.35	2,644.67	2,614.50	2,538.60	2,337.79	2,657.39	2,845.13	2,729.65	2,914.91	3,201.70
2017	3,268.00	3,233.00	2,955.50	2,752.50	2,803.50	2,686.00	2,629.50	2,633.00	2,780.50	2,736.00	2,689.00	2,402.63
2018	2,486.50	2,488.00	2,426.50	2,418.00	2,396.00	2,324.00	2,215.00	2,183.50	2,177.50	2,082.00	1,830.50	1,794.50
2019	2,037.00	2,100.50	1,903.50	2,018.50	1,946.50	1,968.00	1,879.00	2,066.50	2,097.00	2,104.00	2,493.50	2,813.00
2020	3,013.50	2,714.50										

Figure 4.20: Visualization Displaying MPOB Monthly Average CPO Prices for Past Five Years.

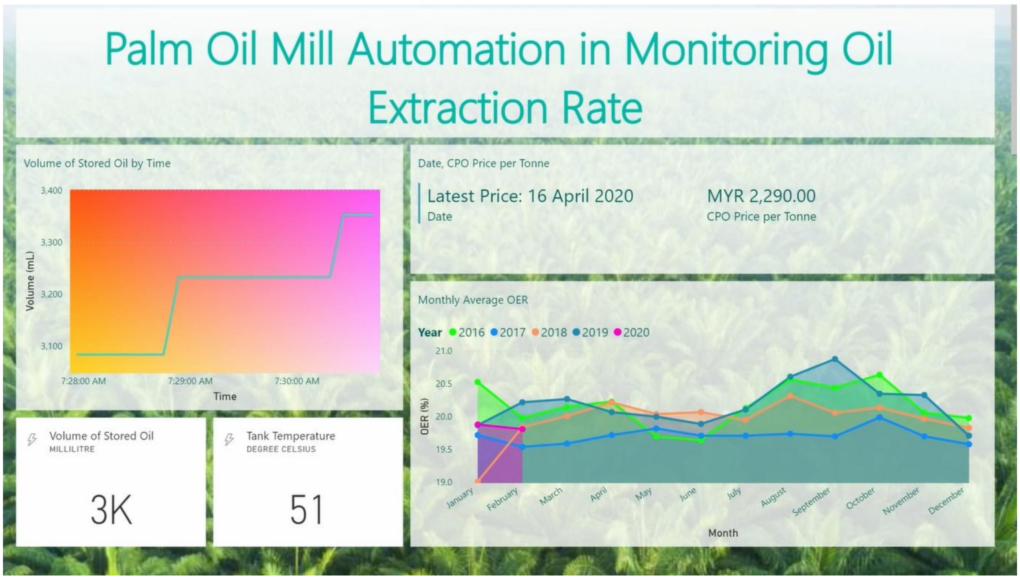


Figure 4.21: Power BI Dashboard showing real-time data streaming of stored oil volume and tank temperature.

4.4.2 Impact of Specific Gravity and Temperature Factor

4.4.2.1 Error in Volume Measurement

The error in volume measurement was resulted from the difference between the value of apparent volume and actual volume of stored oil in the tank. The actual oil volume was calculated with essential consideration of the effect of temperature of the storage tank, which were denoted by the specific gravity of palm oil changing with each degree Celsius and the temperature factor of tank due to heat expansion, in which all the mentioned parameters were ignored for apparent volume calculation. The line and clustered column chart in Figure 4.22 showed the general overestimation of volume of stored oil by the apparent volume measurement. The detailed data of apparent volume, actual volume and volume measurement error by date and time was presented in Table 4.5. The strong impact of specific gravity and temperature factor can be inferred due to the high percentage error ranged from 13.93 % to 14.91 %.

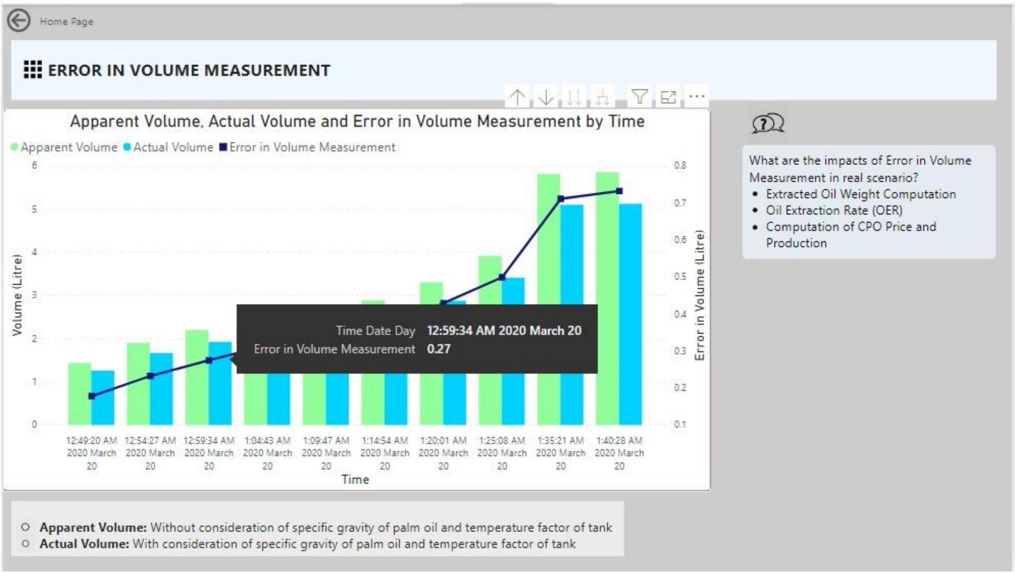


Figure 4.22: Report Page Showing Error in Volume Measurement.

Table 4.5: Apparent Volume, Actual Volume and Error in Volume Measurement by Time.

Date	**Time**	**Apparent Volume (Litre)**	**Actual Volume (Litre)**	**Error in Volume Measurement (Litre)**	**Percentage error (%)**
Friday, March 20, 2020	12:49:20 AM	1.43881	1.26062	0.17819	14.14
Friday, March 20, 2020	12:54:27 AM	1.90246	1.66979	0.23267	13.93
Friday, March 20, 2020	12:59:34 AM	2.20159	1.92669	0.2749	14.27

Friday, March 20, 2020	1:04:43 AM	2.50820	2.19229	0.31591	14.41
Friday, March 20, 2020	1:09:47 AM	2.57550	2.25391	0.32159	14.27
Friday, March 20, 2020	1:14:54 AM	2.88959	2.52887	0.36072	14.26
Friday, March 20, 2020	1:20:01 AM	3.30837	2.87900	0.42937	14.91
Friday, March 20, 2020	1:25:08 AM	3.91410	3.41463	0.49947	14.63
Friday, March 20, 2020	1:35:21 AM	5.81357	5.10212	0.71145	13.94
Friday, March 20, 2020	1:40:28 AM	5.85844	5.12592	0.73252	14.29
Total		**32.41063**	**28.35384**	**4.05679**	**-**

4.4.2.2 Error in Extracted Oil Weight, OER and CPO Production Computation in Real-World Application

While the maximum volume of the tank used in this study model was 10 Litre or equal to 0.01 metric tonne (MT), the palm oil storage tank used in conventional oil mill factory has the maximum capacity of 2000 MT (Malaysian Palm Oil Board, 2010). Hence, the ratio of the model applied in this study to the real-world practice is 1:200000.

By referring to Figure 4.23, the single line and column chart at the left showed the error in volume measurement and OER in real-world scenario whereas the relationship between the error in gain weight of extracted oil and production computation in real-world application was demonstrated in the chart at right. Meanwhile, all three cards in the report page displayed the total value of error in volume measurement, gain weight and production respectively. Presentation of all the relevant data were shown in Table 4.6. The data showed that the error in production was dependant on the error in gain weight of extracted oil. This indicated that all the aforementioned parameters were interrelated as the volume measurement error would cause inaccuracy in gain weight and the OER computation, in which OER as one of the key parameter in computing CPO price for a batch of FFB input, could further affect the production of CPO. Therefore, the impact of consideration of specific gravity and temperature factor into the volume and weight computation becomes significant in real-world application, as concluded from the results obtained.

The current practice of the palm oil mill operators for the computation of the CPO price per tonne is done by using the daily CPO reference price released by MPOB and average OER of mills by regions with the grading of FFB input. FFB is graded by FFB graders and the quality deductions (poor quality penalty) is undertaken as prescribed in the MPOB FFB Grading Manual (Nordin, et al., n.d.). Hence, the OER determination is of the utmost importance to yield the mill production in a more precise way. In this study, the error in OER due to the computation error in volume measurement and extracted oil gain weight, as a result of negligence of the specific gravity and temperature factor, had given rise to a total loss RM 32.99K or 14.20 % from production of extracted oil (RM 232.4K of actual production from the extracted oil with the ratio of 1:20000 applied) to the mill for overestimation of the CPO production of 14.57 tonnes in total.

In the meantime, oil palm plantation owners who have a large area of estate will be beneficial from this system due to the precise measurement of OER for various FFB batches. This could provide useful information for the owner to measure the profit per area indirectly as a part of precision harvest monitoring. In current conventional practice of yield per hectare, there are incidents where those “underperforming” ill or unhealthy trees might be covered up by the whole average, hiding the chances to make improvement on the yield quality with further analysis on the data. In addition, most of the individual oil palm smallholders would sell their FFB through dealers, who play an intermediary role between the smallholders and the mill owners, rather than selling them directly to the mills due to small amount of FFB produced or transportation cost. Therefore, smallholders face difficulty in determining their actual FFB production with a couple of service charges imposed by dealers. With the OER monitoring system, smallholders could assess their respective

FFB yield performance and gain the profit from dealers based on individual smallholders instead of overall sales.

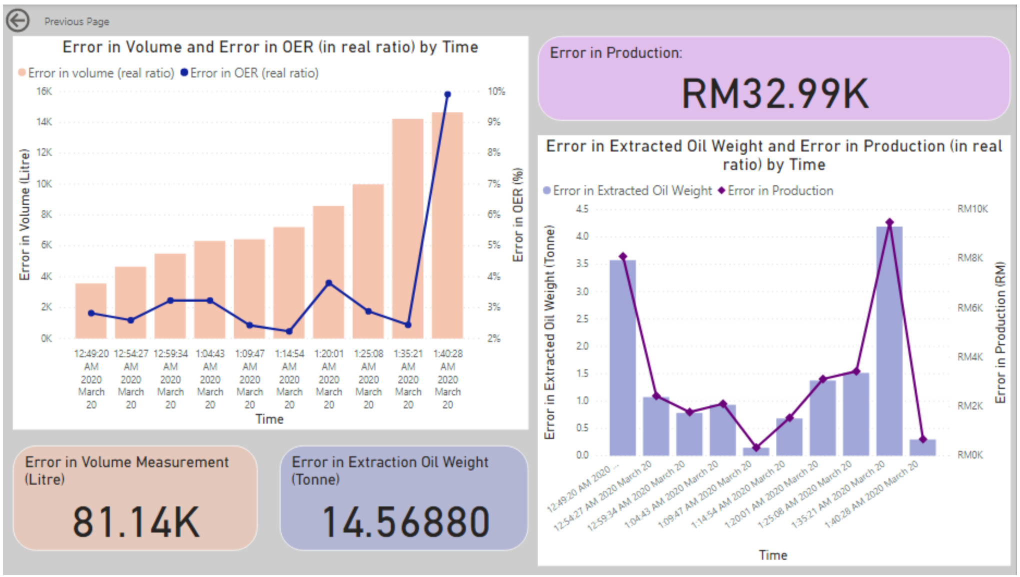


Figure 4.23: Visualization Displaying Error in Volume, OER, Gain Weight of Extracted Oil and CPO Production by Time in Real-world Application.

Table 4.6: Error in Volume Measurement, Gain Weight of Extracted Oil, OER and CPO Production in Real-world Scenario.

Date	Time	In Real-world Scenario			
		Error in Volume Measurement (Litre)	**Error in Gain Weight of Extracted Oil (MT)**	**Error in OER (%)**	**Error in Production (RM)**
Friday, March 20, 2020	12:49:20 AM	3563.60452	3.5762	2.82	8,098.30
Friday, March 20, 2020	12:54:27 AM	4653.25180	1.0730	2.59	2,429.81
Friday, March 20, 2020	12:59:34 AM	5497.97786	0.7826	3.23	1,772.20
Friday, March 20, 2020	1:04:43 AM	6318.18640	0.9322	3.23	2,110.97
Friday, March 20, 2020	1:09:47 AM	6431.73434	0.1460	2.43	330.62

Friday, March 20, 2020	1:14:54 AM	7214.36774	0.6818	2.23	1,543.94
Friday, March 20, 2020	1:20:01 AM	8587.32026	1.3756	3.80	3,115.05
Friday, March 20, 2020	1:25:08 AM	9989.28420	1.5146	2.88	3,429.81
Friday, March 20, 2020	1:35:21 AM	14228.9426	4.1894	2.44	9,486.90
Friday, March 20, 2020	1:40:28 AM	14650.38496	0.2974	9.91	673.46
Total		**81135.05468**	**14.5688**	**-**	**32,991.76**

4.5 Summary

In summary, the result of the experiment have proven that the hardware and software implementation of the cloud-based OER monitoring system is feasible. Sensor data were uploaded from Arduino UNO to ThingSpeak cloud platform via ESP8266 Wifi module. The Microsoft Power BI then acquired the data input from ThingSpeak website to visualize and analyse the data at user interface. The real-time data streaming of oil volume and tank temperature were displayed in Power BI dashboard. The result established had emphasized the significant impact of computation of gain weight of extracted oil with consideration of oil specific gravity and tank temperature factor. The OER monitoring system could bring about significant financial impact to palm oil millers as the error in oil volume measurement of about 94.96K litre with extracted oil weight of 14.57 metric tonnes in total could surprisingly lead to a total loss RM 31.33K or 14.20 % from the extracted oil production to the mill. The visualization of data in Power BI report was impactful and user-friendly.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

To conclude this project, the IoT based palm oil mill automation in oil extraction rate (OER) monitoring system prototype was successfully developed and tested. The system was made IoT based to automate and establish connectivity among mill processes, increasing productivity and efficiency. The input data from sensor node were uploaded to ThingSpeak IoT platform and then to Microsoft Power BI to perform further data analysis and visualizations. The display of real-time streaming of tank storage condition and temperature on Power BI dashboard allows the user to monitor the oil tank storage condition which help in curtailing the theft of crude palm oil in oil mill. It is worth to mention that in this study the computation of volume and weight of extracted oil with the purity of palm oil, tank volume expansion and density properties of oil taken into consideration have established a marked effect to the OER as well as the palm oil production, which was shown by the experiment result of 14.20% of production loss to the mill. This is beneficial to mill operators with reduction of production loss and theft prevention as well as to plantation owners to assess FFB yield and identify underperforming trees. With the system proposed in this project, the millers will strive to improve their productivity by further enhancing the management and efficiency of the mill as well as controlling the quality of the FFB received. Meanwhile, the smallholders will be encouraged to deliver good quality of FFB to the mill.

5.2 Recommendations for future work

There are some improvements and enhancements that are recommended to be applied in this project. Data alert can be done in Power BI streaming dataset which will send notifications to the user when the condition or threshold set for the tracked data is reached or fulfilled. Nonetheless, the data alert setting on the dashboard tiles is only available for Power BI account with Pro license. Thus for future improvement, user can set the data alert for abnormal decrease in volume or tank temperature threshold. In addition, other open source cloud computing platforms can be used, such as Microsoft Azure Cloud services which provides more sophisticated services and functions to enable flexibility in customer experience.

Additionally, the hardware implementation of the prototype can be improved by producing a single-sided printed circuit board (PCB) with all the electronic components being soldered and connected on a single board. PCB can simplify the circuit connection and reduce the overall board weight, providing mechanical support to the electrical components apart from electrical connection.

Moreover, the data obtained for different batches of FFB with the curve of calculated OER can be used for further analysis related to the factors affecting OER such as weather conditions to serve the information for the plantation management team to manage the harvesting process efficiently. Production forecast can be done with the past records of OER data acquired over time as an important input into the decision-making process by the oil mill operators.

Furthermore, it is recommended to consider and combine the input from plantation context such as the quantitative monitoring of number of FFB harvested. With the input data of FFB number combining with the OER monitoring system, enabling trees as a smaller production unit rather than the number of hectares can be made possible by the overall system which is more sophisticated. Precision farming with Profit per Tree (PPT) measurement can be achieved by detection and location of oil palm FFB harvested in plantation. Machine learning algorithm can be applied to acquire the quantity of FFB harvested with the detection of FFB dropping sound using acoustic sensor.

REFERENCES

Andrew, Y.K.C., n.d. Clarification tank simulation. *Palm Oil Engineering Buletin*, [online] Available at: < http://palmoilis.mpob.gov.my/publications/POEB/poeb79-1.pdf> [Accessed 9 June 2019].

Aruna, P., 2018. CPO prices drop further. *The Star Online*, [online] 13 November. Available at: <https://www.thestar.com.my/business/business-news/2018/11/13/cpo-prices-drop-further/> [Accessed 28 July 2019].

Ashwini, P., Pavan, S. and Roja, R.A., 2018. A comparative study of Arduino Uno, NodeMCU in IoT. *International Journal of Scientific Research in Science, Engineering and Technology*, 4(7), pp. 7-11.

Bansal, R.K., 2008. *A textbook of fluid mechanics.* New Delhi: Laxmi Publications (P) Ltd.

Bird, J., 2014. *Understanding engineering mathematics*. New York: Routledge.

Brumbi, D., 2002. *Measuring process and storage tank level with radar technology.* In: International Radar Conference. Alexandria, USA, 8-11 May 1995. Duisburg, Germany: IEEE.

Case, R.P., White Consolidated Industries, Inc., 1980. *Electronic liquid level monitor and controller.* U.S. Pat. 4,201,240.

Chang, L.C., Sani, A.R.A. and Basran, Z., 2003. An economic perspective of oil extraction rate in the oil palm industry of Malaysia. *Oil Palm Industry Economic Journal*, 3(1), pp. 25-31.

Charles, A., Ayylusamy, C., Deepak, R.J., Kumar, K. and Thiruvarasan, K.E., 2018. IoT based water level monitoring system using LabVIEW. *International Journal of Pure and Applied Mathematics*, 118(20), pp. 9-13.

Crawford, C.D., King, J.A. and Randolph, J.H., King J.A. & Company, Inc, Major Oil Company, Inc., 1992. *System for monitoring storage tanks.* U.S. Pat. 5,132,923.

Cytron, 2013. *Product user's manual – HC-SR04 ultrasonic sensor.* [online] Available at: <https://datasheetspdf.com/pdf-file/1291829/Cytron/HC-SR04/1> [Accessed 7 August 2019].

Davies, R.M., 2016. Effect of the temperature on dynamic viscosity, density and flow rate of some vegetable oils. *Journal of Scientific Research in Engineering & Technology*, 1(1), pp. 14-22.

DFRobot, n.d. *SRF02 ultrasonic sensor SKU SEN0005.* [online] Available at: <https://wiki.dfrobot.com/SRF02_Ultrasonic_sensor__SKU_SEN0005_> [Accessed 7 August 2019].

Dulphy-Vigor, V., Ferenczi, L. and Viard, N., 2002. *Device for monitoring the storage level of a product stored in a tank.* U.S. Pat. 6,345,214.

Edwards, J.E. and Otterson, D.W., 2014. Tech Talk: (3) Applying liquid level measurement. *Measurement and Control*, 47(5), pp. 153-157.

Endress and Hauser, 2017. *Continuous level measurement in liquids and bulk solids: Selection and engineering guide for the process industry*. [online] Available at: <https://portal.endress.com/wa001/dla/5000492/2837/000/05/CP00023F00EN1817.pdf> [Accessed 7 August 2019].

Faisal, M.R., 2018. *Crude palm oil monitoring system in vertical tank with ultrasonic sensor using neural network.* Degree. Universitas Sumatera Utara. Available at: <http://repositori.usu.ac.id/bitstream/handle/123456789/4041/111402017.pdf?sequence=1&isAllowed=y> [Accessed 6 August 2019].

Ferretti, M.D., Gabel, B.L., Horton, J.A, Weiss, T.S. and Miller, K.K., Air Products and Chemicals, Inc., 1988. *Method and system for measurement of a liquid level in a tank.* U.S. Pat. 4,788,648.

Gama-Moreno, L.A., Corralejo, A, Ramirez-Molina, A., Torres- Rangel, J.A., Martinez-Hernandez, C. and Juarez, M.A., 2016. A design of a water tanks monitoring system based on mobile devices. *2016 International Conference on Mechatronics, Electronics and Automotive Engineering*, pp. 133-138.

GRobotronics, n.d. *Temperature Sensor - Waterproof (LM35)*. [image]. Available at: <https://grobotronics.com/lm35.html?sl=en> [Accessed 11 August 2019].

Hafiz, M., Renjani, R.A., Haryanto, A., Araswati, N. and Subrata, I.D.M., 2016. *Design of temperature and volume control system at crude palm oil (CPO) storage tank.* In: Department of Mechanical and Biosystem Engineering, Bogor Agricultural University, The 1st International Conference on the Role of Agricultural Engineering for Sustainable Agriculture Production. Indonesia, 13-14 December 2016. Bogor, Indonesia: Research Gate.

Halliday, D., Resnick, R. and Walker, J., 2017. *Fundamentals of physics.* [e-book] Hoboken: John Wiley & Sons. Available at: Google Books <https://books.google.com> [Accessed 12 August 2019].

Hughes, J.C. and Keysar, B.C., 1963. *Testing of metal volumetric standards.* [e-book] Washington: National Bureau of Standards. Available at: Google Books <https://books.google.com> [Accessed 20 January 2019].

Husni, M., Siahaan, H.T., Ciptaningtyas, H.T., Studiawan, H. and Aliarham, Y.P., 2016. Liquid volume monitoring based on ultrasonic sensor and Arduino microcontroller. *International Conference on Innovation in Engineering and Vocational Education*, 128, pp. 1-9.

Joy-it, 2018. *NodeMCU ESP8266 microcontroller development board.* [online] Available at: <https://asset.conrad.com/media10/add/160267/c1/-/en/001774899DS01/datasheet-1774899-programming-language-textbook-das-esp8266-praxisbuch-mit-nodemcu-und-esplorer-nodemcu-card-erik-bartmann.pdf> [Accessed 11 August 2020].

Kauffman, R., n.d. *Oil change monitoring system.* Ohio, United States: University of Dayton Office of Technology Partnerships.

LIPICO Technologies, 2008. *Technical references - Palm oil properties.* [online] Available at: <http://www.lipico.com/technical_references_palm_oil_properties.html> [Accessed 16 January 2020].

Loke, E., 2013. Reduced oil extraction rate for palm oil causing smallholders sleepless night. *The Star Online*, [online] 18 July. Available at: <https://www.thestar.com.my/news/community/2013/07/18/trying-times-for-smallholders-reduced-oil-extraction-rate-for-palm-oil-causing-them-sleepless-night/> [Accessed 28 July 2019].

Mahindar, R., Prakash, M., Ghosh, S., Mukherjee, S. and Ghosh D.R., 2018. IoT-based home appliances control system using NodeMCU and Blynk Server. *International Advanced Research Journal in Science, Engineering and Technology*, 5(6), pp. 16-22.

Mai, S. and Barjenbruch, U., 2016. *Water level measurements with radar gauges at the German North Sea Coast.* Koblenz, Germany: German Federal Institute of Hydrology.

Malaysian Palm Oil Board, 2010. *Recommended practices for storage and transport of edible oils and fats*. Washington: MPOBTAS.

Malaysian Palm Oil Board, 2020. *Oil extraction rate.* [online] Available at: < http://bepi.mpob.gov.my/index.php/en/statistics/oil-extraction-rate.html> [Accessed 23 March 2020].

Malaysian Palm Oil Board, 2020. *Daily CPO prices.* [online] Available at: <http://mpob.gov.my/> [Accessed 19 March 2020].

Malaysian Palm Oil Board, 2020. *MPOB daily Malaysia prices of crude palm oil (RM/Tonne).* [online] Available at: < http://bepi.mpob.gov.my/admin2/price_local_daily_view_cpo_msia.php?more=Y&jenis=1Y&tahun=2020> [Accessed 19 March 2020].

Malaysian Palm Oil Board, 2020. *Oil extraction rate.* [online] Available at: < http://bepi.mpob.gov.my/index.php/en/?option=com_content&view=category &id=111> [Accessed 19 March 2020].

Motzer, J., 2000. *A pulse radar gauge for level measurement and process control.* In: 2000 IEEE MTT-S International Microwave Symposium Digest. Boston, USA, 11-16 June 2000. Schiltach, Germany: IEEE.

Nordin, J., Chandramohan, D., Basiron, Y. and Venugopal, R., n.d. MPOB daily FFB reference price: Towards a more transparent market. *Oil Palm Industry Economic Journal*, 1, pp. 17-20.

Parallax, 2013. *PING))) ultrasonic distance sensor (#28015)*. [online] Available at: <https://www.parallax.com/sites/default/files/downloads/28015-PING-Sensor-Product-Guide-v2.0.pdf > [Accessed 7 August 2019].

Robot-R-Us, n.d. *Stainless steel waterproof temperature sensor*. [online] Available at: <https://www.robot-r-us.com/vmchk/sensor-temp/humid/one-wire-waterproof-temperature-sensor.html> [Accessed 11 August 2019].

RS Components, n.d. *Arduino UNO*. [online] Available at: < https://docs.rs-online.com/ab83/0900766b80e8ba21.pdf> [Accessed 11 August 2019].

Sarip, M.S.M., Morad, N.A., Yamashita, Y., Tsuji, T., Yunus, M.A.C., Aziz, M.K.A. and Lam, H.L., 2016. Crude palm oil (CPO) extraction using hot compressed water (HCW). *Separation and Purification Technology*, [e-journal], 169. Available through: Universiti Tunku Abdul Rahman Library website <http://library.utar.edu.my/> [Accessed 2 June 2019].

Smith, G.E., Quintex Research International, Inc., 1984. *Specific gravity independent gauging of liquid filled tanks.* U.S. Pat. 4,446,730.

Suda, S. and Keisch, B., n.d. *Temperature effects and corrections in volume measurements based on liquid-level detection.* Upton, New York: Brookhaven National Laboratory.

Surya, S. and Chauhan, S.S., 2015. Water level indicator with temperature sensor. *IOSR Journal of Electrical and Electronics Engineering*, 10(3), pp. 65-71.

Suthergreen, D.B., Cotton, F.W. and Zingel, R.M., Optimum Applied Systems, Inc., 1994. *System for monitoring fills of liquid in a tank container and ongoing liquid condition.* U.S. Pat. 5,351,725.

Texas Instrument, 2017. *LM35 Precision centigrade temperature sensors.* [online] Available at: <http://www.ti.com/lit/ds/symlink/lm35.pdf> [Accessed 11 August 2019].

Yahya, S.I., 2013. A multi-level storage tank gauging and monitoring system using a nanosecond pulse. *International Journal of Engineering Trends and Technology (IJETT)*, 5(1), pp. 17-24.

Zulkefli, F., Othman, N., Syahlan, S., Zaini, M.R. and Bakar, M.A., 2017. Fresh fruit bunch quality and oil losses in milling processes as factors that affect the extraction rate of palm oil. *International Journal of Agriculture, Forestry and Plantation*, 5, pp. 99-103.

APPENDICES

APPENDIX A: Datasheet

1.0 INTRODUCTION

The HC-SR04 ultrasonic sensor uses sonar to determine distance to an object like bats or dolphins do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package. From 2cm to 400 cm or 1" to 13 feet. It operation is not affected by sunlight or black material like Sharp rangefinders are (although acoustically soft materials like cloth can be difficult to detect). It comes complete with ultrasonic transmitter and receiver module.

Features:

- Power Supply :+5V DC
- Quiescent Current : <2mA
- Working Currnt: 15mA
- Effectual Angle: <15°
- Ranging Distance : 2cm – 400 cm/1" - 13ft
- Resolution : 0.3 cm
- Measuring Angle: 30 degree
- Trigger Input Pulse width: 10uS
- Dimension: 45mm x 20mm x 15mm

DatasheetA-1: Technical Features of HC-SR04

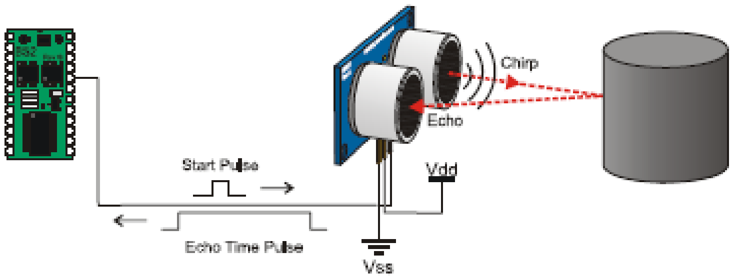
Web Site: www.parallax.com
Forums: forums.parallax.com
Sales: sales@parallax.com
Technical: support@parallax.com

Office: (916) 624-8333
Fax: (916) 624-8003
Sales: (888) 512-1024
Tech Support: (888) 997-8267

PING))) Ultrasonic Distance Sensor (#28015)

The Parallax PING)))™ ultrasonic distance sensor provides precise, non-contact distance measurements from about 2 cm (0.8 inches) to 3 meters (3.3 yards). It is very easy to connect to microcontrollers such as the BASIC Stamp®, Propeller chip, or Arduino, requiring only one I/O pin.

The PING))) sensor works by transmitting an ultrasonic (well above human hearing range) burst and providing an output pulse that corresponds to the time required for the burst echo to return to the sensor. By measuring the echo pulse width, the distance to target can easily be calculated.



Features

- Range: 2 cm to 3 m (0.8 in to 3.3 yd)
- Burst indicator LED shows sensor activity
- Bidirectional TTL pulse interface on a single I/O pin can communicate with 5 V TTL or 3.3 V CMOS microcontrollers
- Input trigger: positive TTL pulse, 2 µs min, 5 µs typ.
- Echo pulse: positive TTL pulse, 115 µs minimum to 18.5 ms maximum.
- RoHS Compliant

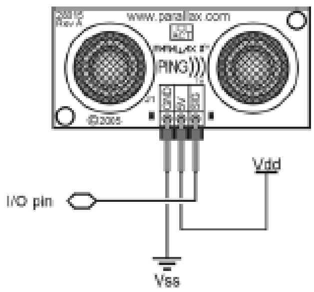
Key Specifications

- Supply voltage: +5 VDC
- Supply current: 30 mA typ; 35 mA max
- Communication: Positive TTL pulse
- Package: 3-pin SIP, 0.1" spacing (ground, power, signal)
- Operating temperature: 0 – 70° C.
- Size: 22 mm H x 46 mm W x 16 mm D (0.84 in x 1.8 in x 0.6 in)
- Weight: 9 g (0.32 oz)

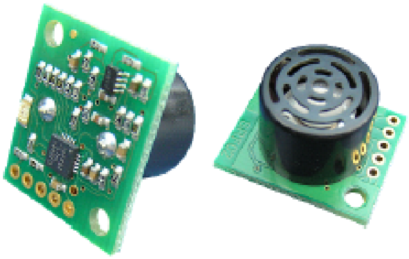
Pin Definitions

GND	Ground (Vss)
5 V	5 VDC (Vdd)
SIG	Signal (I/O pin)

The PING))) sensor has a male 3-pin header used to supply ground, power (+5 VDC) and signal. The header may be plugged into a directly into solderless breadboard, or into a standard 3-wire extension cable (Parallax part #800-00120).



DatasheetA-2: Features and Key Specifications of Parallax PING))) Ultrasonic Distance Sensor.



(https://www.dfrobot.com/product-325.html)

Introduction

The SRF02 is a single transducer ultrasonic rangefinder in a small footprint PCB. It features both I2C and a Serial interfaces. The serial interface is a standard TTL level UART format at 9600 baud,1 start, 2 stop and no parity bits, and may be connected directly to the serial ports on any microcontroller. Up to 16 SRF02's may be connected together on a single bus, either I2C or Serial. New commands in the SRF02 include the ability to send an ultrasonic burst on its own without a reception cycle, and the ability to perform a reception cycle without the preceding burst. This has been as requested feature on our sonar's and the SRF02 is the first to see its implementation. Because the SRF02 uses a single transducer for both transmission and reception, the minimum range is higher than our other dual transducer rangers. The minimum measurement range varies from around 17-18cm (7 inches) on a warm day down to around 15-16cm (6 inches) on a cool day. Like all our rangefinders, the SRF02 can measure in uS, cm or inches. Reference Links: http://www.robot-electronics.co.uk/htm/srf02tech.htm (http://www.robot-electronics.co.uk/htm/srf02tech.htm%3Cbr%3E);

Specification

- Range: 16cm to 6m.
- Power: 5v, 4mA Typ.
- Frequency: 40KHz.
- Size: 24mm x 20mm x 17mm height.
- Analogue Gain: Automatic 64 step gain control
- Connection Modes: 1 - Standard I2C Bus 2 - Serial Bus(connects up to 16 devices to any uP or UART serial port)
- Full Automatic Tuning: No calibration, just power up and go
- Timing: Fully timed echo, freeing host controller of task.
- Units: Range reported in uS, mm or inches.

DatasheetA-3: Introduction and Specifications of SRF02 ultrasonic sensor SKU SEN0005.

4/20/2020 Stainless Steel Waterproof Temperature Sensor | Sensor - Temp/Humid | Robot R Us

View Full-Size Image

Stainless Steel Waterproof Temperature Sensor

(E)

SG$3.50

This is a waterproofed version of the DS18B20 Temperature sensor. Handy for when you need to measure something far away, or in wet conditions. While the sensor is good up to 125°C the cable is jacketed in PVC so we suggest keeping it under 100°C. Because they are digital, you don't get any signal degradation even over long distances! The DS18B20 provides 9 to 12-bit (configurable) temperature readings over a 1-Wire interface, so that only one wire (and ground) needs to be connected from a central microprocessor.Usable with 3.0-5.5V systems.

Because each DS18B20 contains a unique silicon serial number, multiple DS18B20s can exist on the same 1-Wire bus. This allows for placing temperature sensors in many different places. Applications where this feature is useful include HVAC environmental controls, sensing temperatures inside buildings,equipment or machinery, and process monitoring and control.

Features

9 to 12 bit selectable resolution.
one digital pin for communication.
Multiple sensors can share one pin.
Query time is less than 750ms.

Specification

3.0V to 5.5V input.
Temperature range: -55 to 125°C (-67°F to +257°F).
±0.5°C Accuracy from -10°C to +85°C.
Red wire - VCC.
Black wire - GND.
Yellow wire - DATA.
Stainless steel tube 6mm diameter by 30mm long.
Cable length: 90cm.

Usage

Here, we will show how to use the waterproof temperature sensor (DS18B20) with your Arduino. This sensor uses the one wire protocol to talk with the microcontroller. So, it requires only one digital port to communicate.

Hardware

The sensor has 3 wires: red (VCC), black (GND) and white (DATA). Connect the red to +5V, the black to GND and the white to the digital pin D10. Then, put a 4.7kohm resistor between the white wire and the +5V.

https://www.robot-r-us.com/vmchk/sensor-temp/humid/index2.php?option=com_virtuemart&Itemid=50&category_id=36&flypage=flypage_new.tpl... 1/2

DatasheetA-4: Features, Specifications and Usage of DS18B20 Temperature Sensor.

Product Folder | Order Now | Technical Documents | Tools & Software | Support & Community

TEXAS INSTRUMENTS

LM35

SNIS159H – AUGUST 1999 – REVISED DECEMBER 2017

LM35 Precision Centigrade Temperature Sensors

1 Features

- Calibrated Directly in Celsius (Centigrade)
- Linear + 10-mV/°C Scale Factor
- 0.5°C Ensured Accuracy (at 25°C)
- Rated for Full −55°C to 150°C Range
- Suitable for Remote Applications
- Low-Cost Due to Wafer-Level Trimming
- Operates From 4 V to 30 V
- Less Than 60-μA Current Drain
- Low Self-Heating, 0.08°C in Still Air
- Non-Linearity Only ±¼°C Typical
- Low-Impedance Output, 0.1 Ω for 1-mA Load

2 Applications

- Power Supplies
- Battery Management
- HVAC
- Appliances

3 Description

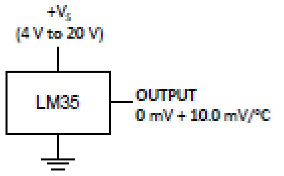
The LM35 series are precision integrated-circuit temperature devices with an output voltage linearly-proportional to the Centigrade temperature. The LM35 device has an advantage over linear temperature sensors calibrated in Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling. The LM35 device does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full −55°C to 150°C temperature range. Lower cost is assured by trimming and calibration at the wafer level. The low-output impedance, linear output, and precise inherent calibration of the LM35 device makes interfacing to readout or control circuitry especially easy. The device is used with single power supplies, or with plus and minus supplies. As the LM35 device draws only 60 μA from the supply, it has very low self-heating of less than 0.1°C in still air. The LM35 device is rated to operate over a −55°C to 150°C temperature range, while the LM35C device is rated for a −40°C to 110°C range (−10° with improved accuracy). The LM35-series devices are available packaged in hermetic TO transistor packages, while the LM35C, LM35CA, and LM35D devices are available in the plastic TO-92 transistor package. The LM35D device is available in an 8-lead surface-mount small-outline package and a plastic TO-220 package.

Device Information[1]

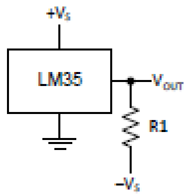
PART NUMBER	PACKAGE	BODY SIZE (NOM)
LM35	TO-CAN (3)	4.699 mm × 4.699 mm
	TO-92 (3)	4.30 mm × 4.30 mm
	SOIC (8)	4.90 mm × 3.91 mm
	TO-220 (3)	14.986 mm × 10.16 mm

(1) For all available packages, see the orderable addendum at the end of the datasheet.

Basic Centigrade Temperature Sensor (2°C to 150°C)



Full-Range Centigrade Temperature Sensor



Choose $R_1 = -V_S / 50\ \mu A$
V_{OUT} = 1500 mV at 150°C
V_{OUT} = 250 mV at 25°C
V_{OUT} = −550 mV at −55°C

DatasheetA-5: Features, Applications and Description of LM35 Precision Centigrade Temperature Sensors.

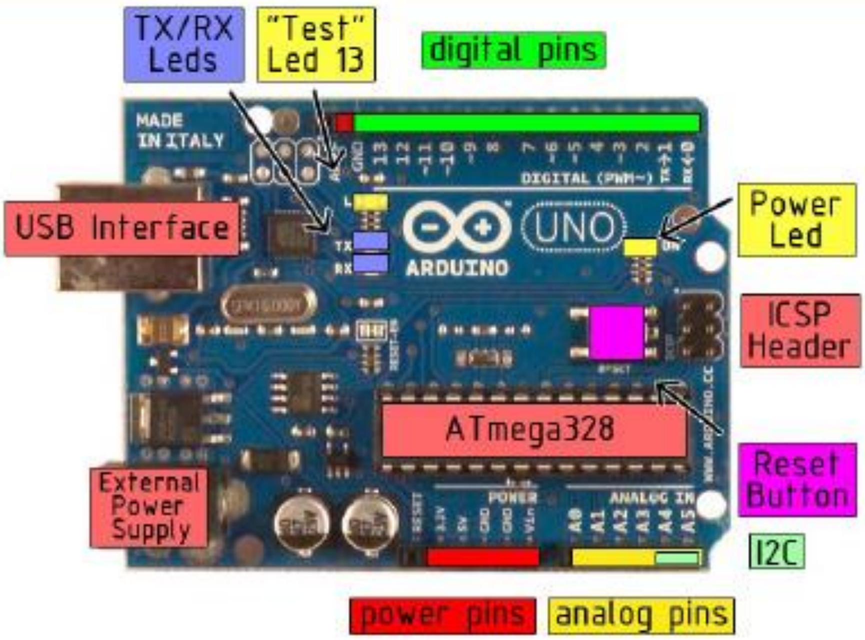
Technical Specification

EAGLE files: arduino-duemilanove-uno-design.zip Schematic: arduino-uno-schematic.pdf

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

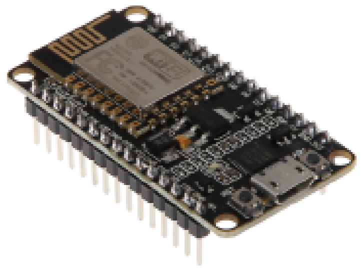
the board



DatasheetA-6: Technical Specification of Arduino UNO ATmega328 Microcontroller Development Board.

NodeMCU ESP8266

Microcontroller Development Board



Technical Specifications

Model	NodeMCU ESP8266
Article No.	SBC-NodeMCU
Type	ESP8266
Processor	Tensilica LX106
Clock Frequency	80 - 160 MHz
RAM	64 kB
Data Storage	96 kB
Wireless Standard	802.11 b/g/n
Frequency	2.4 GHz
Data Interfaces	UART / I2C / PWM / GPIO
Operating Voltage	3.0 - 3.6 V (operable via 5V-microUSB)
Operating Current	80mA
Operating Temperature	-40°C - 125°C
Dimensions (W x D x H)	58 x 31 x 12 mm
Scope Of Delivery	NodeMCU ESP8266
EAN	4250236815923

Published: 16.03.2018 Time for more www.joy-it.net

DatasheetA-7: Technical Specifications of NodeMCU ESP8266 Microcontroller Development Board.

APPENDIX B: System Program Code

Program CodeB-1: Program Code in Arduino IDE for Data Upload to Thingspeak

```
#include <SoftwareSerial.h>          //Software Serial library
SoftwareSerial espSerial(10, 11);    //Pin 10 and 11 act as RX
and TX. Connect them to TX and RX of ESP8266
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
#include <DallasTemperature.h>
#include <math.h>

#define DEBUG true
String mySSID = "Jessica's";            // WiFi SSID
String myPWD = "2468abcd";              // WiFi Password
String myAPI = "UED1X5A5Q9J5G6Y3";      // API Key
String myHOSTID = "184.106.153.149";    //ThingSpeak IP address
String myPORT = "80";

// Data wire is plugged into digital pin 2 on the Arduino
#define ONE_WIRE_BUS 2
// Setup a oneWire instance to communicate with any OneWire
device
OneWire oneWire(ONE_WIRE_BUS);
// Pass oneWire reference to DallasTemperature library
DallasTemperature sensors(&oneWire);

#define pi 3.1416
#define beta_tank 0.00005

const int sensor_pin = 7;
float level;
float avg_temp;
float volume;
float SG;
float TF;
float weight_MT;
```

```
void setup()
{
  Serial.begin(9600);
  espSerial.begin(9600);
  pinMode(sensor_pin,INPUT);
  sensors.begin();

  lcd.begin(16,2);
  lcd.clear();

  espData("AT+RST", 1000, DEBUG);     //Reset the ESP8266 module
  espData("AT+CWMODE=1", 1000, DEBUG);
  espData("AT+CWJAP=\""+ mySSID +"\",\""+ myPWD +"\"", 1000,
DEBUG);   //Connect to WiFi network
  Serial.print("Wait for connection...");
  while(!espSerial.find("OK"))
  {
      Serial.print(".");                //Wait for connection
  }
  delay(1000);
}

void loop()
{
    level = cal_level();
    avg_temp = cal_temp(level);
    volume = cal_volume(level);
    SG = cal_SpecificGravity(avg_temp);
    TF = cal_TemperatureFactor(avg_temp);
    weight_MT = cal_weight (volume, SG, TF);

    String sendData = "GET /update?api_key="+ myAPI
+"&field1="+String(level,2)+"&field2="+String(avg_temp,2)+"&fie
ld3="+String(volume,5)+"&field4="+String(SG,5)+"&field5="+Strin
g(TF,5)+"&field6="+String(weight_MT,5);
    espData("AT+CIPMUX=1", 1000, DEBUG);
    espData("AT+CIPSTART=0,\"TCP\",\""+ myHOSTID +"\","+
myPORT, 1000, DEBUG);
    espData("AT+CIPSEND=0,"
+String(sendData.length()+4),1000,DEBUG);
    espSerial.find(">");
```

```
    espSerial.println(sendData);
    Serial.println(sendData);
    Serial.println("Value to be sent: ");
    Serial.print("Level :");
    Serial.print(level);
    Serial.println(" cm");
    Serial.print("Average temperature :");
    Serial.print(avg_temp);
    Serial.write(0xC2);       //shows degrees character
    Serial.write(0xB0);
    Serial.println("C");
    Serial.print("Volume :");
    Serial.print(volume);
    Serial.println(" L");
    Serial.print("SG     :");
    Serial.println(SG);
    Serial.print("TF     :");
    Serial.println(TF);
    Serial.print("Weight :");
    Serial.print(weight_MT);
    Serial.println(" MT");

    lcd.setCursor(0,0);
    lcd.print("LEVEL   : ");
    lcd.setCursor(9,0);
    lcd.print(float(level));
    lcd.print("cm ");
    lcd.setCursor(0,1);
    lcd.print("VOLUME  : ");
    lcd.setCursor(9,1);
    lcd.print(float(volume));
    lcd.print("L ");

    espData("AT+CIPCLOSE=0",1000,DEBUG);
    delay(5000);
  }

float cal_level()
{
  float duration,cm;
```

```
  pinMode(sensor_pin, OUTPUT);
  digitalWrite(sensor_pin, LOW);
  delayMicroseconds(2);
  digitalWrite(sensor_pin, HIGH);
  delayMicroseconds(5);
  digitalWrite(sensor_pin, LOW);

  pinMode(sensor_pin, INPUT);
  duration = pulseIn(sensor_pin, HIGH);

  cm = 28.8- microsecondsToCentimeters(duration);

  return cm;
}

float microsecondsToCentimeters(float microseconds)
{
  return microseconds/29/2;
}

float cal_temp(float level)
{
  float tempC_1, tempC_2;
  int i=0;

  // Send command to all the sensors for temperature conversion
  sensors.requestTemperatures();

  // Display temperature from each sensor
  Serial.print("Temp 1: ");   //Temp1 is from upper temp sensor
  tempC_1 = sensors.getTempCByIndex(0);
  Serial.print(tempC_1);
  Serial.write(0xC2);         //shows degrees character
  Serial.write(0xB0);
  Serial.print("C  |  ");
  Serial.print(DallasTemperature::toFahrenheit(tempC_1));
  Serial.write(0xC2);         //shows degrees character
  Serial.write(0xB0);
  Serial.println("F");

  Serial.print("Temp 2: ");   //Temp2 is from lower temp sensor
```

```
  tempC_2 = sensors.getTempCByIndex(1);
  Serial.print(tempC_2);
  Serial.write(0xC2);         //shows degrees character
  Serial.write(0xB0);
  Serial.print("C  |  ");
  Serial.print(DallasTemperature::toFahrenheit(tempC_2));
  Serial.write(0xC2);         //shows degrees character
  Serial.write(0xB0);
  Serial.println("F");
  if (level>14.25)            //if oil level is over the half
of the tank where the upper temp sensor is placed
  {
    avg_temp = (tempC_1+tempC_2)/2;     //get average
temperature of two sensors
  }else                  //if oil level is below half of the tank
  {
    avg_temp= tempC_2;        //get temperature of lower temp
sensor
  }

  Serial.print("Average temperature: ");
  Serial.print(avg_temp);
  Serial.write(0xC2);         //shows degrees character
  Serial.write(0xB0);
  Serial.println("C");
  Serial.println("");

  delay(1000);

  return avg_temp;
}

float cal_volume(float level)
{
  volume = (pi * pow((0.235/2),2) * (level/100))*1000;
//tank diameter=0.235; 1 litre=0.001 cubic meter
  return volume;
}

float cal_SpecificGravity(float avg_temp)
{
```

```
  SG = (-0.0005)*avg_temp + 0.8995;
  return SG;
}

float cal_TemperatureFactor(float avg_temp)
{
  TF = (beta_tank*(avg_temp-25)) + 1;
  return TF;
}

float cal_weight (float volume, float SG, float TF)
{
  float weight_kg;
  weight_kg = volume*SG*TF;
  weight_MT = weight_kg*0.001;
  return weight_MT;
}

String espData(String command, const int timeout, boolean
debug)
{
  Serial.print("AT Command ==> ");
  Serial.print(command);
  Serial.println("     ");

  String response = "";
  espSerial.println(command);
  long int time = millis();
  while ( (time + timeout) > millis())
  {
    while (espSerial.available())
    {
      char c = espSerial.read();
      response += c;
    }
  }

  return response;
}
```

Program CodeB-2: Python Script for Power BI Real-time Data Streaming of Volume of Stored Oil and Tank Temperature from ThingSpeak Channel

```
import urllib.request
import requests
import threading
import json
import pandas as pd
from datetime import datetime
from datetime import timedelta
import time
import random
import urllib

def read_volume_data_thingspeak():
  URL='https://api.thingspeak.com/channels/1040964/fields/3.json?api
_key='
  KEY='LKA2V856BBJNC6EA  '
  HEADER='&results=1'
  VOL_URL=URL+KEY+HEADER

  get_vol_data=requests.get(VOL_URL).json()

  global vol

  for item in  get_vol_data['feeds']:
    for k,v in item.items():
      vol = item['field3']

  Volume=int (float(vol)*1000)

  return Volume
```

```
def read_temp_data_thingspeak():
    URL='https://api.thingspeak.com/channels/1040964/fields/2.json?api
_key='
    KEY='LKA2V856BBJNC6EA '
    HEADER='&results=1'
    TEMP_URL=URL+KEY+HEADER

    get_temp_data=requests.get(TEMP_URL).json()

    global temp

    for item in  get_temp_data['feeds']:
        for k,v in item.items():
            temp = item['field2']

    Temp=int (float(temp))

    return Temp

def data_generation():
    Date = datetime.today().strftime("%Y-%m-%d")
    Time = datetime.now().isoformat()
    Volume=read_volume_data_thingspeak()
    Temperature=read_temp_data_thingspeak()

    return [Date, Time,Volume, Temperature]

if __name__ == '__main__':
    REST_API_URL = 'https://api.powerbi.com/beta/4edf9354-0b3b-429a-
bb8f-f21f957f1d1c/datasets/b42ed2c2-0b9e-457a-9054-93025202cd12
/rows?ctid=4edf9354-0b3b-429a-bb8f-f21f957f1d1c&key=maIHToWks9
bCU3iSHd0kjZmcKQ1Fcinc3TTv6X8xMVM5lJf8u%2BTdVUEQUh7XnPaTR
lR%2B6yZlJsNLsRoZTs1ofg%3D%3D'
```

```
while True:
    data_raw = []
    for i in range(1):
        row = data_generation()
        data_raw.append(row)
        print("Raw data - ", data_raw)
    # set the header record
    HEADER = ["Date", "Time","Volume","Temperature"]

    data_df = pd.DataFrame(data_raw, columns=HEADER)
    data_json = bytes(data_df.to_json(orient='records'), encoding='utf-8')
    print("JSON dataset", data_json)

    # Post the data on the Power BI API
    req = requests.post(REST_API_URL, data_json)

    print("Data posted in Power BI API")
    time.sleep(3)
```