DEVEPLOMENT OF DIY OXIMETER FOR HEALTH MONITORING

PROJEK SARJANA MUDA 1 (PSM 1)

ENGKU AIMAN ZAKWAN BIN ENGKU ZAIN

TS. DR NURUL AZMA BINTI ZAKARIA

UNIVERSITI TEKNIKAL MALAYSIA MELAKA (UTeM)

[DEVELOPMENT OF DIY OXIMETER FOR HEALTH MONITORING]

[ENGKU AIMAN ZAKWAN BIN ENGKU ZAIN]

This report is submitted in partial fulfilment of the requirements for the

Bachelor of [Computer Science (Computer Networking)] with Honours.

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY UNIVERSITI TEKNIKAL MALAYSIA MELAKA

[2021]

DECLARATION

I hereby declare that this project report entitled

[DEVELOPMENT OF DIY OXIMETER FOR HEALTH MONITORING]

is written by me and is my own effort and that no part has been plagiarized

without citations.

STUDENT : ________________________________ Date : ________

([ENGKU AIMAN ZAKWAN BIN ENGKU ZAIN])

I hereby declare that I have read this project report and found

this project report is sufficient in term of the scope and quality for the award of

Bachelor of [Computer Science (Computer Networking)] with Honours.

SUPERVISOR : Nurul Azma Zakaria Date : 11/9/2021

([DR. NURUL AZMA BINTI ZAKARIA])

DEDICATION

This project is dedicated to Allah Almighty, my creator, pillar, source of inspiration, wisdom, knowledge, and understanding. Through this project, He has been my source of power, and I have only been able to fly on His wings. I also dedicate this work to my parents, Engku Zain Bin Engku Azam and Engku Fauziah Binti Tuan Jalai, my family, sisters, and brother who has encouraged me all the way and whose encouragement has made sure that. I give it all it takes to finish that which I have started. My supervisor, Dr. Nurul Azma Binti Zakaria, and fellow friends are willing to share and guide me along the journey. Thank you. I can never quantify my love for you. God bless you.

ACKNOWLEDGEMENTS

First and foremost, I would like to show my gratitude to all the fantastic individuals who have helped and encouraged me to ensure my initiatives' success. My supervisor, Dr. Nurul Azma Binti Zakaria, deserves special recognition for her diligent and comprehensive supervision throughout the project's completion. For this endeavor, his profound views have brought numerous inspirations and enlightenment.

Next, I would like to offer my heartfelt gratitude to my wonderful family. I am grateful for their unwavering support during this journey. Their support and care gave me confidence and a good attitude while finishing this project. Gratitude also extends to my students, faculty seniors, and anyone who has volunteered to assist me with their skills.

Not to mention, I would like to express my gratitude to my university, Universiti Teknikal Malaysia Melaka (UTeM), for allowing me to put my talents and talents to the test and give me such a fantastic opportunity to show my worth.

ABSTRACT

A worldwide health crisis that causes life loss in a few nations across the globe, the Covid-19 pandemic. The Covid-19 pandemic reported a broad variety of symptoms, including as shortness of breath and other symptoms. Necessary action is needed to preserve more lives in this nation or globe. This project involves designing and developing a prototype for easy, succinct, and efficient oxygen saturation and heart rate monitoring utilizing the MAX30100 internet-based sensor (IoT). Specifically, the MAX30100 sensor is designed to measure oxygen saturation and heart rate in the percentage value. C, C++, JavaScript may be used to interface with NodeMCU Microcontroller. Due to the current pulse oximeter, the concept to create this project is very costly, feature restriction, not user-friendly. To finish this project, many actions must be taken as a project process. First, the literature study (prior research) needs to be done to obtain the hardware, software, language, technique utilized in the project. Next, specify the components to utilize and create the hardware and software project. In this stage, the components and design must be suitable to fulfil this project's need to answer the issue statement and achieve the project goal. Next, hardware development starts with the implementation of all software and hardware. Finally, this project's testing and verification will be tested, and the project recorded for verification.

ABSTRAK

Krisis kesihatan di seluruh dunia yang menyebabkan kehilangan nyawa di beberapa negara di seluruh dunia, pandemi Covid-19. Pandemik Covid-19 melaporkan pelbagai gejala, termasuk sesak nafas dan gejala lain. Tindakan yang diperlukan diperlukan untuk memelihara lebih banyak nyawa di negara atau dunia ini. Projek ini melibatkan merancang dan mengembangkan prototaip untuk ketepuan oksigen yang mudah, ringkas dan cekap dan pemantauan degupan jantung menggunakan sensor berasaskan internet MAOT30100 (IoT). Secara khusus, sensor MAX30100 dirancang untuk mengukur ketepuan oksigen dan denyut jantung dalam nilai peratusan. C, C ++, JavaScript boleh digunakan untuk berinteraksi dengan NodeMCU Microcontroller. Oleh kerana pulse oximeter semasa, konsep untuk membuat projek ini sangat mahal, sekatan ciri, tidak mesra pengguna. Untuk menyelesaikan projek ini, banyak tindakan mesti dilakukan sebagai proses projek. Pertama, kajian literatur (penyelidikan sebelumnya) perlu dilakukan untuk mendapatkan perkakasan, perisian, bahasa, teknik yang digunakan dalam projek. Seterusnya, tentukan komponen yang akan digunakan dan dibuat projek perkakasan dan perisian. Pada peringkat ini, komponen dan reka bentuk mestilah sesuai untuk memenuhi keperluan projek ini untuk menjawab pernyataan masalah dan mencapai matlamat projek. Seterusnya, pengembangan perkakasan dimulakan dengan pelaksanaan semua perisian dan perkakasan. Akhirnya, ujian dan pengesahan projek ini akan diuji dan projek direkodkan untuk pengesahan.

TABLE OF CONTENTS

LIST OF TABLES

PAGE

LIST OF FIGURES

PAGE

LIST OF ABBREVIATIONS

IoT	**-**	**Internet of Things**
FYP	**-**	**Final Year Project**
BPM	**-**	**Beat Per Minute**
SpO2	**-**	**Oxygen Saturation**
PS	**-**	**Problem Statement**
PQ	**-**	**Project Question**
PO	**-**	**Project Objective**
PC	**-**	**Project Contribution**
FDA	**-**	**U.S. Food and Drug Administration**
MDA	**-**	**Malaysia Device Authority**
SUS	**-**	**System Usability Scale**
IDE	**-**	**Integrated Development Environment**
DIY	**-**	**Do It Yourself**
SDLC	**-**	**Software Development Life Cycle**

LIST OF ATTACHMENTS

CHAPTER 1: INTRODUCTION

1.1 Introduction

A global health crisis that causes life loss and disrupts economies in a few countries worldwide, the Covid-19 pandemic. People infected with the Covid-19 pandemic reported various symptoms, such as respiratory shortness and other signs. In the exchange of oxygen and carbon dioxide, the breathing system is the organ and other parts involved in breathing. The severely affected patients of COVID-19 require ventilators to survive and meet the oxygen requirement. This project also will monitor people's health.

The DIY Oximeter is an IoT project designed to detect people's heart rate and oxygen saturation because people monitor their health with a laptop and phone. DIY Oximeter will be cheaper compared to the commercial oximeter and smartwatch. It notifies or alerts people when oxygen saturation is below that 94 per cent (Dr Guleria, 2021) that can be an excellent way to check their health. DIY Oximeter also can show and record alert data of previous oxygen saturation. This project is expected to help people who have possibly been infected with Covid-19 by given the alerts to the people and will record the last monitoring health when the oxygen saturation is low because prevention is better than cure. Other than that, this system will reduce the number of cases that involve Covid-19 of a people.

1.2 Problem Statement

People may not be sensitive to their health because some people may not be able to buy the commercial oximeter and smartwatch that has been built up to check oxygen saturation in people's bodies. People also may not be sure if their body is healthy because sometimes the oximeter only shows the oxygen saturation data but does not give the alert when their SpO2 is not in good condition. The current oximeter may not save the data because it builds with hardware only, not including the software. To summarize this, the problem statements for this project are shown in Table 1.1.

Table 1.1 Summary of Problem Statement

PS	Problem Statement
PS_1	The market price of a commercial pulse oximeter is expensive to purchase.
PS_2	Users are unaware of their oxygen saturation condition in the home and not be able to see the previous data.
PS_3	Users will not get an alert.

1.3 Project Question

They used project research questions to identify questions on monitoring oxygen saturation and heart rate. Based on several studies, it can be concluded that there are some difficulties in determining whether oxygen saturation or Sp2O is good or not. Table 1.2 shows the summary of the project question.

Table 1.2: Summary of Project Question

PS	PQ	Project Question
PS_1	PQ_1	How can you build a low-cost pulse oximeter that gives reliable readings?
PS_2	PQ_2	How to determine oxygen saturation condition based on Sp2O or heart rate?

PS_3	PQ_3	How do you build a pulse oximeter that alerts you when your oxygen saturation is low?

1.4 Project Objective

Project objectives are tailored to the problem statement and project questions. The objectives of this project are shown in the Table 1.3 below. For this project there are 3 objectives need to be achieved:

PO1: To develop low-cost hardware that can provide accurate readings.

NodeMCU and sensor are simple to get by in the market, and they are also inexpensive, costing between RM 7 and RM 20 to purchase. To guarantee that the readings obtained are correct, the sensors utilized are likewise of high quality.

PO2: To develop software application that can display, and record for people oxygen saturation and heart rate.

To complete prototype that can integrated to the software. The data of parameter will display on the software and can record the data for the user. User can take an action after the data display on the software.

PO3: To develop a system that can provide a user alert.

Because it is essential to observe the user's health level, the system must be able to capture real-time data from the sensor, and this system must give an alarm if the user's health level is in danger.

Table 1.3: Summary of Project Objective

PS	PQ	PO	Project Objective
$\mathbf{PS_1}$ $\mathbf{PS_2}$ $\mathbf{PS_3}$	$\mathbf{PQ_1}$ $\mathbf{PQ_2}$ $\mathbf{PQ_3}$	PO_1	To develop low-cost hardware that can provide accurate readings.
		PO_2	To develop software application that can display, and record for people oxygen saturation.
		PO_3	To develop a system that can provide a user alert.

1.5 Project Scope

This project will be focused on three features. The first feature about detecting oxygen saturation and heart rate are by using the MAX 30100 to see the Sp2O and heart rate of the user. The second feature is to improve the system to meet a required specification to differ between regular or abnormal oxygen saturation and heart rate. The third feature about the alerting system, using a mobile phone or web application, will be sent immediately to the user about the level of Sp2O.

1.6 Project Contribution

Project contribution defines the expected output from this project. This project can help many users to detect one of the symptoms of Covid-19 by monitoring oxygen saturation. The main goal of this project is to monitor the oxygen saturation and heart rate level to avoid the Covid-19 or respiratory system and develop a cheap product.

The utilization of wireless technology eases the project development in monitoring and alerting the mobile phone and detecting parameters of the oxygen saturation and heart rate help take users' lives and save the lives of the users from the respiratory system. The project contribution can be referring to the Table 1.4 below.

Table 1.4: Summary of Project Contribution

PS	PQ	PO	PC	Project Contribution
PS_1, PS_2, PS_3	PQ_1, PQ_2, PQ_3	PO_1	PC_1	Provide advice on how to make low-cost hardware that can be used to get reliable data readings.
		PO_2	PC_2	Provide a mobile application for monitoring and recording sensor data.
		PO_3	PC_3	Propose a real-time record for this project and provide an algorithm for differentiating between low and typical user Sp2O circumstances.

1.7 Report Organization

Chapter 1: Introduction

This chapter discuss about the purpose in developing the Development of DIY Oximeter for Health Monitoring which includes project background, problem statement, project question, and project objective to clarify the intention of the system.

Chapter 2: Literature Review

This chapter discuss about other topics that have similar fields and can correlated to this project. Previous work or related work that using different tools and methods been compared. In this chapter will make changes to the existing work and identify the improvements needed in this project.

Chapter 3: Project Methodology

This chapter discuss the method used to perform this project whether using SDLC model or OOAD model. Every stage of methodology will be recorded in this chapter. Milestones of this project will be determined in this chapter to make well-plan time to complete a certain task or phase of this project.

Chapter 4: Analysis and Design

This chapter will cover all the requirement that need to be analyzed for the project which include the Data Requirement, Functional Requirement, and Non-functional requirement. The results of the analysis and detailed design will be described. This chapter can be considered important where it is necessary to formulate a good analysis and design to produce a successful project.

Chapter 5: Implementation

In this chapter contain the implementation of the project design that has been determined previously. The system development begins using the required software and hardware. The implementation status and progress for each of the component or module been described.

Chapter 6: Testing

Throughout this chapter, the testing will be made based on test plan which are test organization, test environment and test schedule. All the result of the test will be record in this chapter and analyzed. This is where the system will be analyzed and decide whether this project is successful.

Chapter 7: Project Conclusion

In this chapter, conclusion for this project by explaining how the objective has been achieved based on the implantation and testing phase. All the result obtain will be conclude including the weakness and strength of this project.

1.8 Conclusion

In conclusion, this project will improve safety and reduce the respiratory system or Covid-19 cases for users by early detection by using this project. In this chapter, we can identify all objectives, problem statements, project questions, project scope, and project contributions can identify all objectives, problem statements, project questions, project scope, and project contributions in detail. The development of DIY Oximeter Tools for Health Monitoring System can solve the problem users face in monitoring their health, especially the body's oxygen saturation and heart rate. This project can provide an overview of how the sensor method can detect the heart rate and oxygen saturation parameters to be captured and transferred to the mobile phone. Hence, if the process is complete and ready to be used in real-life situations, it can be helpful and provide more ideas in IoT. This project aims in providing a consistent and user-friendly solution to overcome the problem faced by users.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

A literature review is a citation that refers to a topic relevant to the selected research subject. A literature review is a collection of ideas, mathematical methods, and methodological contributions on a particular subject. This chapter will focus on developing a development plan for DIY Oximeter Tools for Health Monitoring. This document includes all the sensor and application descriptions and data. We will also discuss the earlier research on the pulse oximeter in this chapter. It will examine their implementation methods, constraints, and strategies and draw comparisons between the present and previous projects. Many important factors may be used to determine the appropriateness of an existing pulse oximeter for the research material while doing this evaluation. The system or hardware must be built using Internet of Things (IoT) technologies.

2.2 Related Work Background

The domain connected to the Development of DIY Oximeter Tools for Health Monitoring was discussed in a linked study. The goal of this linked project is to find and study a domain that can improve knowledge and comprehension while assisting in the development process.

2.2.1 Background of NodeMCU

The words "node" and "MCU" (microcontroller unit) were merged to form NodeMCU (Yuan, 2017). After the creation of the ESP8266 in 2013, NodeMCU was introduced. The Lua programming language is used by NodeMCU, which is an open-source firmware.

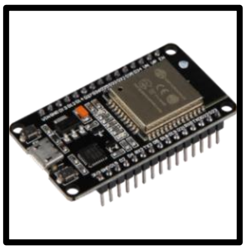


Figure 2.1: ESP8266 Module (Joy-IT, 2021**)**

A dual in-line package (DIP), which comprises a USB controller and a small surface-mounted board, is often used by NodeMCU. It is also Wi-Fi capable, thanks to the Tensilica Xtensa LX106 core. Prototyping on breadboards has never been easier thanks to the DIP format. In 2016, NodeMCU has over 40 modules.

Table 2.1: Specification of ESP32

Model	NodeMCU ESP8266
Type	ESP8266
Processor	Tensilica LX6 Dual-Core
Clock Frequency	240 MHz
SRAM	512 kB
Memory	4 MB
Wireless Standard	802.11 b/g/n
Frequency	2.4 GHz
Bluetooth	Classic/LE
Data Interfaces	UART/I2C/SPI/DAC/ADC
Operating Voltage	3.3V (operable via 5V-microUSB)
Operating Temperature	-40C – 125C
Dimensions	48 x 26 x 11.5mm
Weight	10g

2.2.2 Sensors for Oximetry

The MAX30100 is a heart rate and oxygenation rate sensor. The sensor has two LEDs, a photocoupler, and a CPU designed for analog-to-digital conversions. The voltage range for the chip is 1.8 to 3.3 volts. Its communication protocol is based on the Phillips I2C standard, which is used to connect low-speed peripherals to motherboards, microcontrollers, and other devices (Kalovrektis et al., 2020).

Figure 2.2: Components of MAX30100 Sensor (Pixel Electric**)**

Figure 2.4 shows the MAX30100 sensor's system block diagram, which shows the internal active components for monitoring HbO2 and Hb readings recorded with the photodiode when the Red and IR LEDs are turned on.

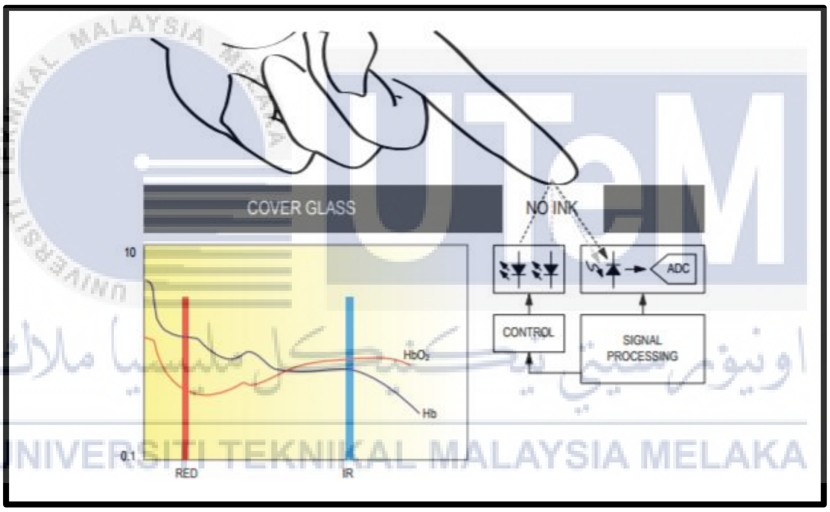


Figure 2.3: Block Diagram of the MAX30100

Figure 2.5 depicts the sensor functional block diagram retrieved from the manufacturer's datasheet to better understand the sensor's internal electrical subsystem. The internal I2C data protocol is used to communicate with and operate the sensor through a wire connection to the IoT embedded platform.

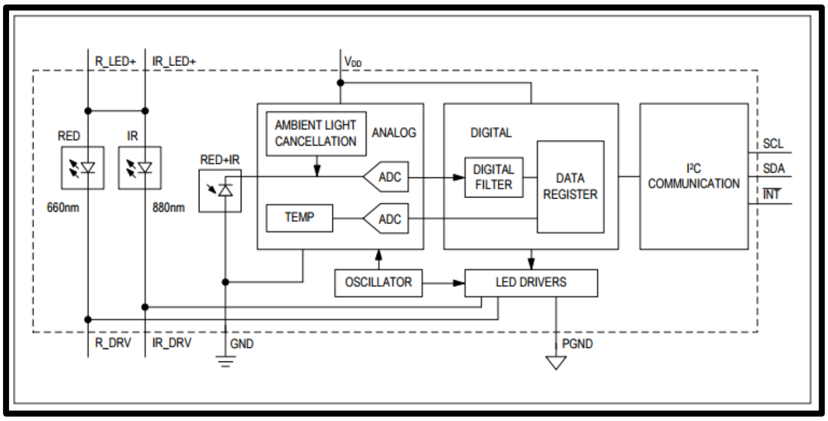


Figure 2.4: Functional Block of the MAX30100 (LEETS ACADEMNY, 2017**)**

2.2.3 Human Requirements of Oxygen Saturation

The percentage of oxygenated haemoglobin in a person's blood transported from the lungs to other bodily organs is called oxygen saturation. This is an essential body survival function. Most COVID-19 patients with severe diseases reported a significant decrease in oxygen levels. While the normal range is 94-100, readings below 94 may induce hypoxemia, leading to several health issues. If your oxygen level goes below 90, it is a sign that your health is failing and it's time to seek medical care.

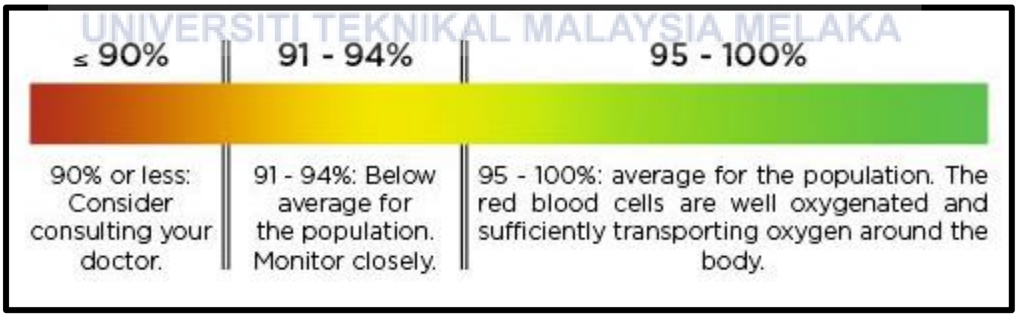


Figure 2.5: Level of Oxygen Saturation (Sp2O) in Human (Pinterest**)**

The percentage of haemoglobin binding sites in oxygen-occupied circulation is assessed by oxygen saturation, often called medicinal "sats." Most haemoglobin is deoxygenated at low partial oxygen. According to an oxygen-haemoglobin dissociation curve, oxygen saturation rises by around 90% (value varies depending on the clinical condition) and reaches 100% at partial oxygen pressures of >11 kPa.

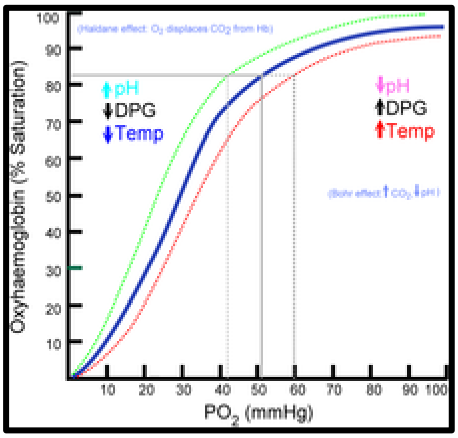


Figure 2.6: Hemoglobin Saturation Curve (Wikipedia, 2021**)**

2.3 Critical Review of Current Problem and Justification

In this section of Chapter 2 Literature Review, several works from previous projects will be compared and analyzed to identify appropriate methodologies, techniques, and components that can be used in this project.

2.3.1 Design of Low-Cost Pulse Oximetry Based on Raspberry Pi

According to (Bakhri, S., Rosiana, E. & Saputra, R. C., 2020), the main aim of this project is to utilize the Raspberry Pi 3 platform for low-cost, reliable oximetry. Now, although the price is lower, the inaccuracy accuracy is comparable to commercial ones. The percentage of total hemoglobin saturation in the blood is measured (SpO2). This research uses wavelengths of red light (660 nm) and infrared light (940 nm) captured by the light sensor after passing through arteries and capillaries at the index fingertip. A finger-driven red and infrared LED is utilized in this study, as well as light passing through a photodiode collected blood vessels. The photodiode signal is transmitted to the Raspberry Pi, where the percentage of blood oxygen saturation is analyzed. The IC MAX30100 sensor alternates between producing red and infrared light, reflected, and captured by photodetector IC. I2C serial communication is used to transmit IC voltage to Arduino. Through a USB connection channel, the Arduino transmits data to the Raspberry Pi (Universal Serial Bus). Arduino provides data as a voltage value from the detected red and infrared light, which IC MAX30100 accepts. The voltage data of Raspberry is then utilized to calculate a blood level of saturation oxygen using the oximetry formula. The value is then displayed on the LCD with a graph of red and infrared light components.

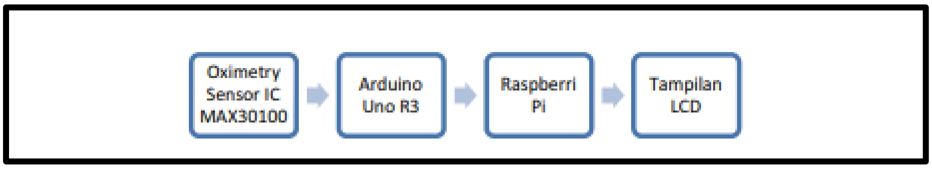


Figure 2.7: Concept Design of Design Low-Cost Oximetry Based on Raspberry Pi (Bakhri, S., Rosiana, E., & Saputra, R. C., 2020**)**

Figure 2.8: Pulse Oximetry made (Bakhri, S., Rosiana, E., & Saputra, R. C., 2020**)**

The aim of the testing and validation phase is to minimize measurement errors from the produced oximetry equipment. Data validation is to evaluate the performance and inaccuracy of the oximeter. (Bakhri, S., Rosiana, E., & Saputra, R. C., 2020) compared the commercial equipment to the pulse oximetry of this study. It may be established that oximetry equals commercially available equipment. The accuracy and error rates observed with certain volunteers are promising.

No	Name	Measurement	Produced Tool SPO₂ (%)	Standard Commercial (%)
1	Rivaldy	1	101,27	100
		2	98,4	100
		3	95,2	100
		4	101,6	99
		5	97,83	100
2	Bambang	1	93,86	98
		2	94,5	97
		3	95,8	96
		4	95,52	95
		5	96,34	96
3	Adit	1	96,79	100
		2	99,05	99
		3	95,59	98
		4	96,49	97
		5	94,98	97
4	Jaya	1	96,87	100

Figure 2.9: Percentage data collection SpO2 between two devices.

2.3.2 A Cost-Effective Pulse Oximeter Designed Response to The Covid-19 Pandemic

The project presents the design of a compact and cost-effective optical pulse oximeter for Royal United Hospital in Bath, UK during the Covid-19 epidemic in 2020, comprising a finger clip, sensor read electronics and related software (Metcalfe, B., Iravani, P., Graham-Harper-Cater, J., Bowman, R., Stirling, J., & Wilson, P., 2021). The device is based on widely used commercial optical sensors that detect reflection of various light wavelengths and are designed for wearables markets. The demand for healthcare practitioners at both national and international levels grew almost intolerable as the number of cases rose dramatically worldwide. The demand for point-of-care SpO2 monitoring for big patient groups led in supply shortages and price increases for commercial sensors.

The display system of Arduino Nano, MAX3010x, and SDD1306 (Metcalfe, B., Iravani, P., Graham-Harper-Cater, J., Bowman, R., Stirling, J., & Wilson, P., 2021) uses three essential components. Using dual wavelength photoplethysmography, this research will monitor human body heart rate and saturation oxygen. MAX30100/MAX30101 is optical and contains required driving and sensor circuitry. The decision to utilize Arduino's platform was focused on keeping things simple. Arduino Nano features a small footprint, built-in power management, and I2C hardware connectors to communicate with a display or sensor. The optical sensor will provide an Arduino Nano heart rate and saturated oxygen data, calculating a time-averaged heart rate and saturated oxygen (SpO2) for display on an OLED screen using the SSD 1306 OLED driver. Due of its high-power density and low cost, a PP3 9V battery was utilized as external power supply. To connect the three components to a few minutes, a simple PCB interposer was utilized.

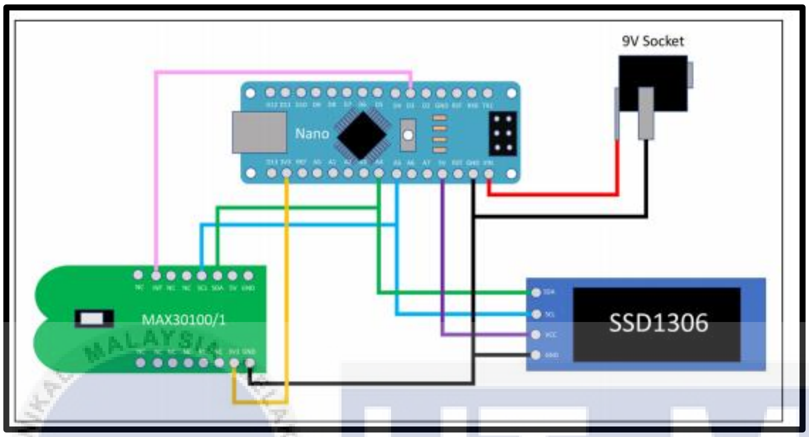


Figure 2.10: System Architecture for Project (Metcalfe, B., Iravani, P., Graham-Harper-Cater, J., Bowman, R., Stirling, J., & Wilson, P., 2021)

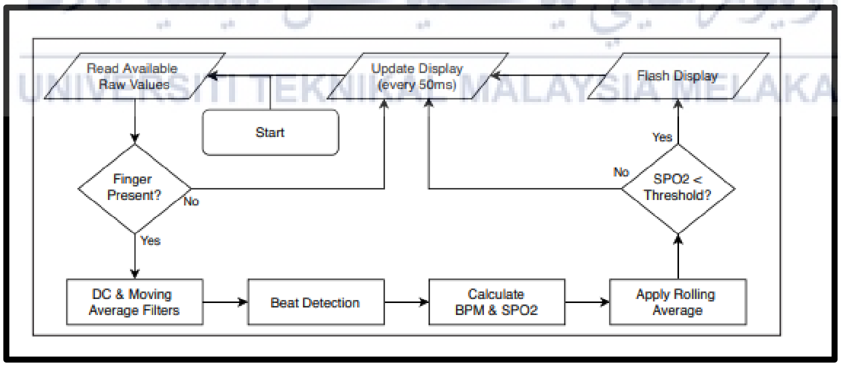


Figure 2.11: Flow Diagram for Project (Metcalfe, B., Iravani, P., Graham-Harper-Cater, J., Bowman, R., Stirling, J., & Wilson, P., 2021)

2.3.3 Design, Simulation, and Implementation of a Digital Pulse Oxygen Saturation Measurement System Using the Arduino Microcontroller

According to (Bhuyan, M. H. & Sarder, M. R., 2021), if a person can routinely check their oxygen saturation level, they may identify their disease early and seek medical treatment. Practically every country in the world identified COVID-19 disease. A pulse oximeter is essential equipment for early detection of such illnesses and is very popular for accurate measurement. Without this device, physicians and health workers cannot detect anomalies in the respiratory system. This project's main objective (Bhuyan, M. H. & Sarder, M. R., 2021) is to provide an automated and low-cost oxygen saturation measurement device that will help and support doctors in monitoring the conditions of their patients. The Arduino microcontroller is a low-cost, widely accessible microcontroller for programming to automate a system.

Arduino Uno R3, Power Supply, MAX30100, and Liquid Crystal Display are the four components of this system. The Arduino Uno will send and receive pulse oximetry data inside and outside the microcontroller, percentage oxygen saturation levels, and send the measured SpO2 level to the LCD. A 5 V power supply unit powers the CPU, pulse oximetry sensor, and LCD unit, which has two lines each with 16 characters. The LCD will then inform the patient about oxygen saturation (SpO2). Finally, the MAX30100 is an integrated pulse oximetry circuit that can monitor heart rate and pulse rate and transmit information to Arduino Uno R3 for percentage calculations.

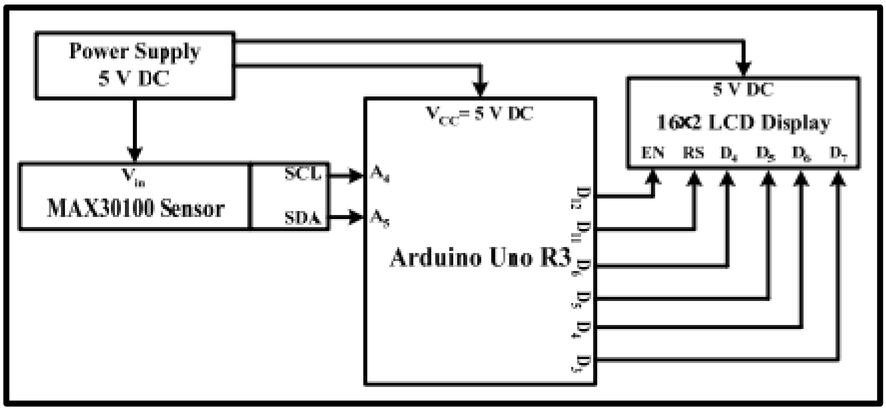


Figure 2.12: Block Diagram for the system (Bhuyan, M. H., & Sarder, M. R., 2021)

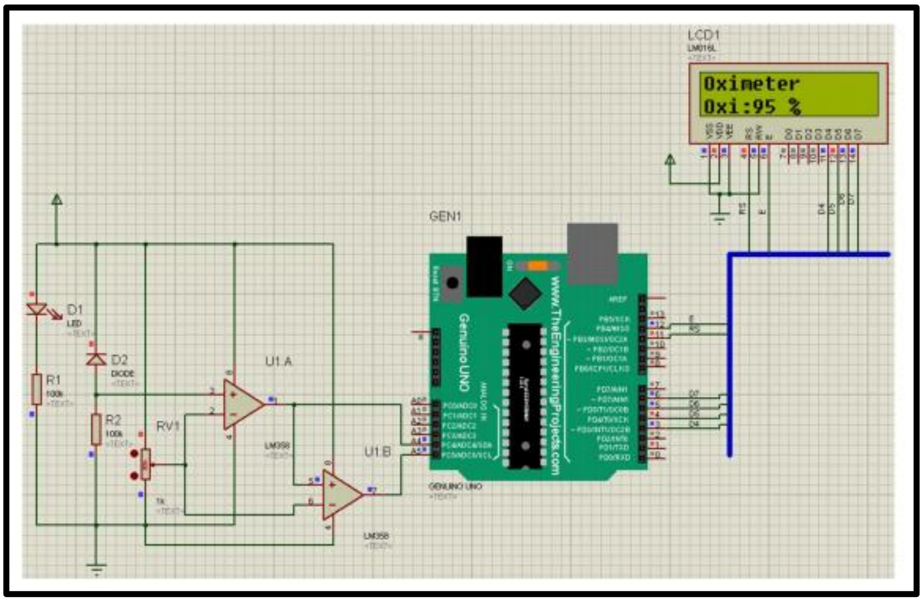


Figure 2.13: Complete Circuit Diagram for the system (Bhuyan, M. H., & Sarder, M. R., 2021)

2.3.4 IoT Pulse Oximetry Status Monitoring for Home Quarantined Covid-19 Patients

We present a prototype experiment for an inexpensive, cordless and internet monitoring system for a Covid-19 infected patient isolated at home in this study (Miron-alexe, V., 2020). The family physician can monitor the patient's pulse and blood oxygen level remotely using a mobile phone, tablet or computer, using an embedded Arduino IoT (Internet of Things) platform and a free online API (Application Programming Interface) live data stream server without any contamination risk or contact with the patient. The account also stores historical data, which may be downloaded for future study as a CSV report.

The proposed pulse oximetry system is based on an RCWL-0530 module with a 1.8V and 3.3V LED sensor (for IR and Red). A person's normal pulse rate is 60-100 beats per minute, and the usual SpO2 level is 95-100%. Figure 2.16 shows the system configuration for this project, including using an ESP32 with Wi-Fi capabilities. The sensor collects the pulse and spo2 values from the patient's finger and then sends them via an internet-connected router to the ThingSpeak API server using an "API write" key and channel number ID to a previously created user account that is periodically monitored by the personal physician on a smartphone, tablet or PC. ThingSpeak's free user account panel is set up and can hold up to 8 chart value fields, enabling many patients to be seen simultaneously. A flowchart in Figure 2.17 streamlines this project's algorithm.

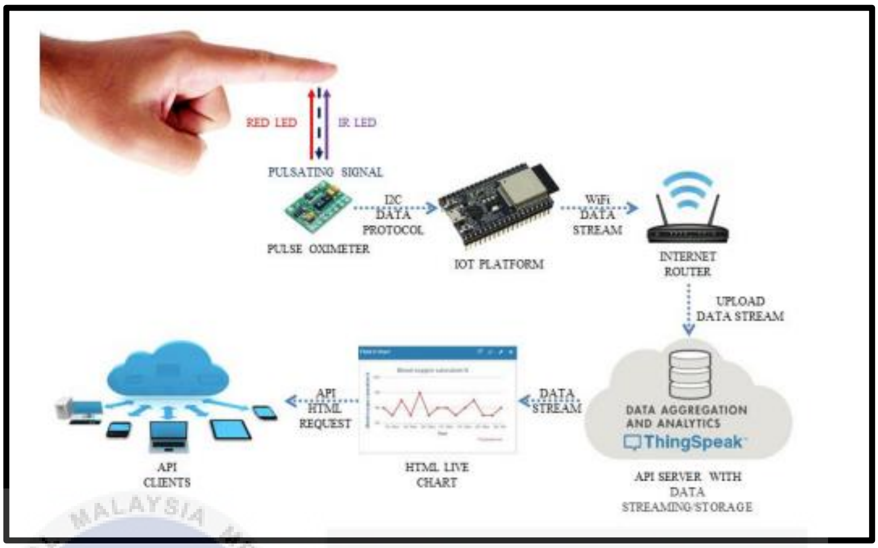


Figure 2.14: Proposed System (Miron-alexe, V., 2020)

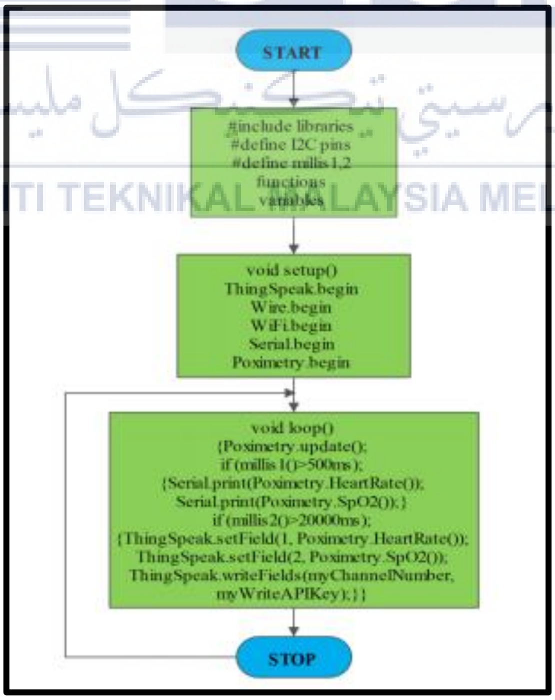


Figure 2.15: Flowchart Algorithm (Miron-alexe, V., 2020)

2.3.5 The Development of Heart Rate Detection Using Arduino Microcontroller

Heart attack has been especially deadly in recent years. This attack cannot be predicted, but it can be recognised by monitoring human pulse, a crucial cardiovascular health sign. Heart rate is the condition of health of the human circulatory system, affected by variables such as workplace stress, pre- or post-sport physical effort, and psychological issues. Unfortunately, some people do not know their heart rate before or after physical exercise. This research offers a heart rate measuring method using a pulse sensor, Arduino microcontroller, and Android smartphone. It works on using a light source and detector to monitor changes in our bodies' blood volume. We also utilise the ECG or EKG (electrocardiogram) waveform to compare the pulse sensor with the ECG waveform to estimate heart rate. An infrared light-emitting diode (LED) and a sensor photodiode. The LED delivers infrared light into the fingertip, reflected in the arteries by blood. The results show that this tool can detect and display heart rate on an Android screen. The sensor detects heart rate value in less than 10 seconds. In normal circumstances, SMS will be delivered to the heart rate (BPM, Beat Per Minute 100).

The hardware and software components of this system are separated. Microcontroller includes the hardware module. The microcontroller controls all hardware component operations. The GSM module acts as a bridge between sensor and smartphone, enabling two-way communication and linking the hardware and software components of the system to Android platform. This system operates when the pulse sensor detects heart rate abnormalities and transmits Arduino Uno information. Arduino Uno will handle heart rate data through pulse sensors. The microcontroller circuit module acts as a media interface for smartphone real-time data transfer.

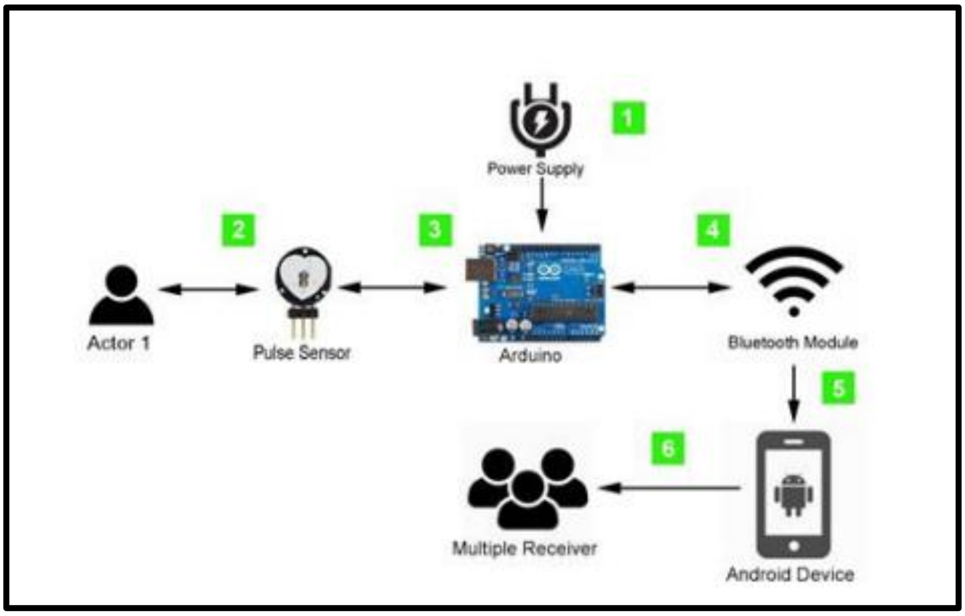


Figure 2.16: Design of Project (Sihombing, P., Barus, Y. E., Sembiring, S., & Zamzami, E. M., 2020)

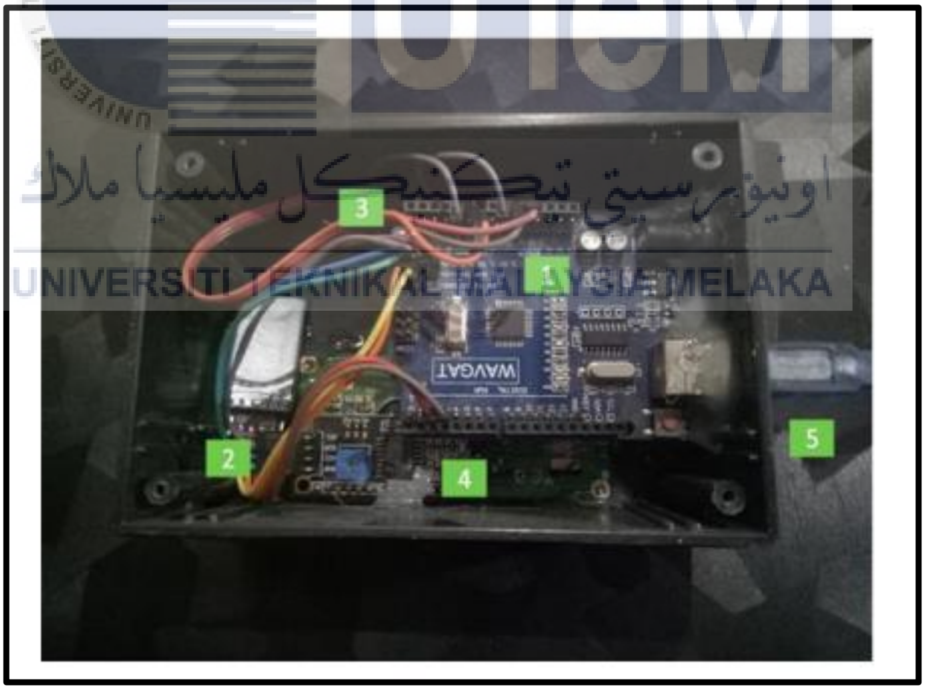


Figure 2.17: Complete Design of Project (Sihombing, P., Barus, Y. E., Sembiring, S., & Zamzami, E. M., 2020)

2.3.6 Severity Monitoring Device for Covid-19 Positive Patients

In this research, we describe the creation of an intelligent wearable device for positive COVID-19 patients, as well as the algorithm that can predict and detect the increase in severity of the virus. Node-MCU (ESP32), MAX 30102 (Pulse Oximeter and Heart Rate Sensor), LM35 (Temperature Sensor) are all utilised in this device. This device will simultaneously monitor and process the patient's physiological condition, such as heart rate, oxygen saturation level, body temperature, and restless hand movements. Therefore, the patient caregiver will be alarmed when the virus is anticipated to progress to the following stage. This device will notify you when the patient's COVID-19 condition develops from mild to moderate.

This research collects patient information such as body temperature, heart rate, oxygen saturation, and hand movement, and analyses everything at the edge using an efficient algorithm to predict the virus' transition from one stage to the next. All data collected and edge computing findings will be sent to the cloud for more efficient processing and reaction, depending on the connectivity availability. Therefore, the patient will be informed depending on the results. Web-based or Bluetooth-based message may convey this warning. An adult's normal respiration rate changes with age; for a resting adult, the median range is 12-20 breaths per minute. On average, the heart beats 60-70 times per minute, whereas breathing happens at around one-fifth of that rate. Typical average body temperature is 98.6 degrees (37 degrees Celsius). According to some studies, body temperature "normal" may range from 97°F (36.1°C) to 99°F (37.2°C). SpO2 is usually at least 95% in a healthy individual.

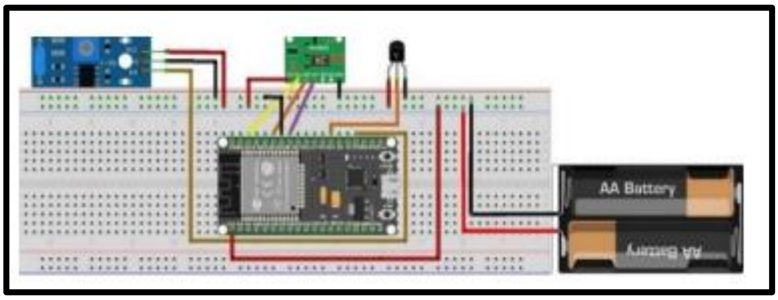


Figure 2.18: Circuit Connection (A. Dhadge and G. Tilekar, 2020)

From the six prior projects on Table 2.2, the author did not indicate which methods they used, but he did say what we should apply for the current project. The authors have utilized a variety of microcontrollers to construct their projects, including the Raspberry Pi, Arduino Nano, Arduino Uno, and NodeMCU, but for the suggested project, I chose the NodeMCU due to its Wi-Fi connection capability. When it comes to Wi-Fi, the range is much greater than that of the Bluetooth Module, making it ideal for Internet of Things applications (IoT). The NodeMCU does not need any additional modules, unlike the Arduino Nano, Raspberry Pi, and Arduino Uno, which require the purchase of a Wi-Fi or Bluetooth module. Many writers continued to utilize the MAX30100 sensor, whereas just one used the MAX30102 to detect their oxygen saturation and heart rate monitor. I choose to purchase the MAX30100 since the author (Bakhri, S., Rosiana, E., & Saputra, R. C., 2020) has tested it and found it to be very useful. The primary reason I chose MAX30100 is because it is simpler to obtain in the market and costs less than RM 6 – 7.

Table 2.2: Comparison of Previous Project

Author, Year	Type of Microcontroller	Type of Sensor	Functionality	Communication Technology	Alert Signal Range	Cost
(S Bakhri, 2020)	Raspberry Pi Display Module	MAX30100	- Checking oxygen saturation and pulse of user. - Send data through display module.	No	No	Rp. 1.163.000 (RM33.50)
(Benjamin Metcalfe, Pejman Iravani, Jonathan-Graham Harper-Cater, Richard Bowman, 2021)	Arduino Nano Display Module	MAX30100	- Checking oxygen saturation and pulse of user. - Send data through display module.	No	No	No-stated
(Muhibul Haque Bhuyan, Md. Refat Sarder, 2021)	Arduino Uno R3 16x2 LCD Display	MAX30100	- Monitor oxygen	No	No	US$36.57

			saturation level intermittently. - Sent to the output port of the microcontroller to display the measured pulse oxygen saturation level data in the text format on the LCD screen of the system.			(RM 147.61)
(Viorel Miron-Alexe, 2020)	ESP32	MAX30100	- Reads of the oxygen saturation and pulse. - Visualize the data through an account to	Wi-Fi	Long	No-stated

			monitor the patient health using Wi-Fi module.			
(Poltak Sihom, 2020)	Arduino Uno Bluetooth Module	Pulse Sensor	- Detecting heart rate of user. - Send alert signal using Bluetooth Module	Bluetooth Module	Short	No-stated
(A. Dhadge and G. Tilekar, 2020)	ESP32	MAX30102 LM35	- Monitor the patient's body condition such as heart pulse rate, oxygen saturation level, body temperature. - Alert will be sent to the	Bluetooth Module	Short	Inexpensive

			person taking care of the patient using Bluetooth.			
Propose Solution	NodeMCU	MAX 301000 (Oxygen Saturation and Heart Rate Sensor)	- Detecting oxygen saturation and pulse of the user. - Sending alert notification using Wi-Fi module to the smartphone - Visualize the data to monitor patient oxygen saturation - Generate report or save	Wi-Fi	Long	Cheap (RM 30 - 40)

			data oximeter of user.			

Table 2.3 chose Dashboard (Visualize Data), Alert, Display Data, Save Data, Communication Link, and Mobile Application for this project. I decided on the dashboard (visualize) because it simplifies monitoring oxygen saturation and heart rate while also attracting users to use this system. Additionally, I chose alerts because some do not execute actions, which is critical if something happens to the user. Apart from displaying and recording data, another objective I am pursuing is alerting. Additionally, I present the data to the user via the resulting phone, and the data is generated into a dashboard. I chose to save data because it is critical for the user to see their current health status if something happens to their oxygen saturation or heart rate. Communication is crucial for an IoT project; I use Wi-Fi for communication because I can use it over long distances. Finally, I use a phone because it is more user-friendly than a computer or web application.

Table 2.3: Functionality Comparison Between Previous Projects

Author, Year	Dashboard (Visualize Data)	Alert	Display Data	Save Data	Communication Link	Mobile Application
(S Bakhri, 2020)	✗	✗	✓	✗	✗	✗
(Benjamin Metcalfe, Pejman Iravani, Jonathan-Graham Harper-Cater, Richard Bowman, 2021)	✗	✗	✓	✗	✗	✗
(Muhibul Haque Bhuyan, Md. Refat Sarder, 2021)	✗	✗	✓	✗	✗	✗
(Viorel Miron-Alexe, 2020)	✓	✗	✓	✓	✓	✗
(Poltak Sihombing, 2020)	✗	✓	✓	✗	✓	✓

(A. Dhadge and G. Tilekar, 2020)	✗	✓	✓	✗	✓	✗
Propose Solution	✓	✓	✓	✓	✓	✓

Ten devices with sensors to measure heart rate and oxygen saturation from five smartwatches and five pulse oximeters with a Food and Drug Administration (FDA) or Medical Device Authority (MDA) certificate are included in Table 2.4. According to the FDA, there are two kinds of pulse oximeters: prescription and over the counter (OTC). Prescription pulse oximeters are FDA-approved and need a prescription and are usually used in clinics or hospitals by physicians. The FDA does not regulate these products, and they are not available without a prescription. May purchase them for personal use in drugstores or online. They are not medical equipment, which means they may be more appropriate for those who engage in sports or aviation activities. According to Table 2.4, the price of a wristwatch plus a commercial pulse oximeter is too costly for specific users. In comparison to the wristwatch and suggested solution, several commercial pulse oximeters lack Bluetooth and Wi-Fi modules. The wristwatch and pulse oximeter do not save prior heart rate and oxygen saturation readings.

Table 2.4: Comparison of Commercial Product

Author, Year (Product)	Specification	Type of Sensor	Functionality	Communication Technology	Alert Signal Range	Price
Honor Band 5, 2019	Processor: Unknown RAM: 384KB Display: AMOLED display	- Blood Oxygen Level - 24*7 Heart Rate Monitoring - Sensor Sleep Monitor	- It has sensors for blood oxygen, 24/7 heart rate monitoring, sleep monitoring, and more. - The sleep monitor automatically identifies six kinds of sleeping issues.	Bluetooth Module	Short	RM 120.00
Honor Watch ES, 2020	Processor: Unknown RAM: Unknown Display: 1.64 inches	Accelerometer, gyro, heart rate, SpO2	- It has GPS and sensors including accelerometer, gyro, heart rate, and SpO2.	Bluetooth Module	Short	RM 497.61

Garmin Vivoactive 4, 2019	Processor: Unknown RAM: Unknown Display: 1.3" (33.0 mm) diameter	Accelerometer, gyro, heart rate, SpO2	- The Pulse Ox sensor1 measures your body's oxygen absorption throughout the day and night. - If your heart rate remains high while you're sleeping, the watch will notify you. It also measures your effort throughout activities.	Bluetooth Module	Short	RM 1,680
Garmin Fenix 5x Plus, 2018	Processor: Unknown RAM: Unknown	Accelerometer, gyro, heart rate, SpO2	- This feature is now available in Garmin's flagship Fenix 5x Plus outdoor recreation	Bluetooth Module	Short	RM 1,984.30

	Display: 1.2" (30.4 mm) diameter		wristwatch, allowing users to monitor their blood SpO2 level at any time, particularly at higher elevations. - It also has multisport GPS with wrist heart rate technology, color TOPO maps with Trendline popularity routing to assist identify the best routes, and Bluetooth earphone connectivity.			
Realme Watch S, 2020	Processor: Unknown	Heart Rate Pulse Ox	- The Realme Watch has 14	Bluetooth Module	Short	RM 369

	RAM: Unknown Display: 3.3cm (1.3”)		sports modes, a real-time heart rate monitor, and the Realme Link app allows you to view your health data on your phone and access additional watch functions.			
Casey, 2021 (Beurer Pulse Oximeter PO-30)	Power: 2 batteries AAA Display: OLED	Heart Rate Pulse Ox	- Easy to use pulse/heart rate calculation. - Calculate arterial oxygen saturation	No stated	No stated	RM 587.00
Casey, 2021 (Yuwell Pulse Oximeter YX302)	Display: OLED Power: 2 batteries AAA	Heart Rate Pulse Ox	- The display shows the battery level, pulse, and oxygen saturation. - It shuts off the pulse oximeter in 8 seconds.	No stated	No stated	RM 138.00

Casey, 2021 (Berry Pulse Oximeter BM100)	Display: LED Power: 2 batteries AAA	Heart Rate Pulse Ox	- Asthma and sleep apnea sufferers may simply insert their finger, toe, or earlobe between the housing of the pulse oximeter and monitor. - The screen displays your SpO2 level and pulse rate for each usage. - Bluetooth link	Bluetooth	Short	RM 187.74
Casey, 2021 (Creative Medical Pulse Oximeter PC-60F)	Display: OLED Power: 2 batteries AAA	Heart Rate Pulse Ox	- 2 direction display with Perfusion Index - Automatic power on/off	No stated	No stated	RM 60.00
Casey, 2021 (Rossmax SB210)	Display: OLED	Heart Rate Pulse Ox	- Instant SpO2, pulse rate, and	Bluetooth	Short	RM 699.00

			artery health measurements in 1 minute - Pulse Strength Indicator			
Propose Solution	NodeMCU	MAX 301000 (Oxygen Saturation and Heart Rate Sensor)	- Detecting oxygen saturation and pulse of the user. - Sending alert notification using Wi-Fi module to the smartphone - Visualize the data to monitor patient oxygen saturation - Generate report or save data oximeter of user.	Wi-Fi	Long	Cheap (RM 30 - 40)

2.4 Proposed Solution

This project aims to develop a system for measuring or monitoring heart rate and oxygen saturation with good detection accuracy. Heart rate and oxygen saturation (SpO2) are measured or monitored to determine the health condition of the human body.

The tracking of cases and overcrowding of beds, as well as the exposure of health workers to the epidemic, are two of the issues that governments confront in serving society. Oximetry is a monitoring parameter that becomes required in a patient-focused study. In the presence of Covid-19 in the patient, the oxygen concentration in the blood changes (Coppo et al., 2020). Thus, this can cause affect human health when the decrease in oxygen rates in the respiratory system. Monitoring DIY Oximeter can be detected by using MAX30100 which will constantly record the heart rate and oxygen saturation (SpO2) value per minute. The importance of accurate heart rate and oxygen saturation values to the preparation of accurate data should not be underestimated by users. When the heart rate and oxygen saturation have been recorded, both data must be delivered to the users. The users can monitor the heart rate and oxygen saturation (SpO2) in the body even when the users are not in the hospital and do not have a commercial oximeter. This would increase the safety of the user's system respiratory.

The system would have a NodeMCU-ESP8266, which will read data from the MAX30100 sensor to detect the oxygen saturation and heart rate from users to get data information then send the data information to a computer or laptop for visualizing the data using open-source Internet of Thing (IoT) platform. Analog-to-Digital Converter is implemented in this system to transform data from analog to digital which can make data information readable. The data from the IoT platform would be submitted to the user's smartphone and the users always online. If the oxygen saturation and heart rate parameters exceed the action level, a warning message will be sent to the mobile. The users will get an alert notification from their mobile and will take the action to seek help from a doctor or professional health.

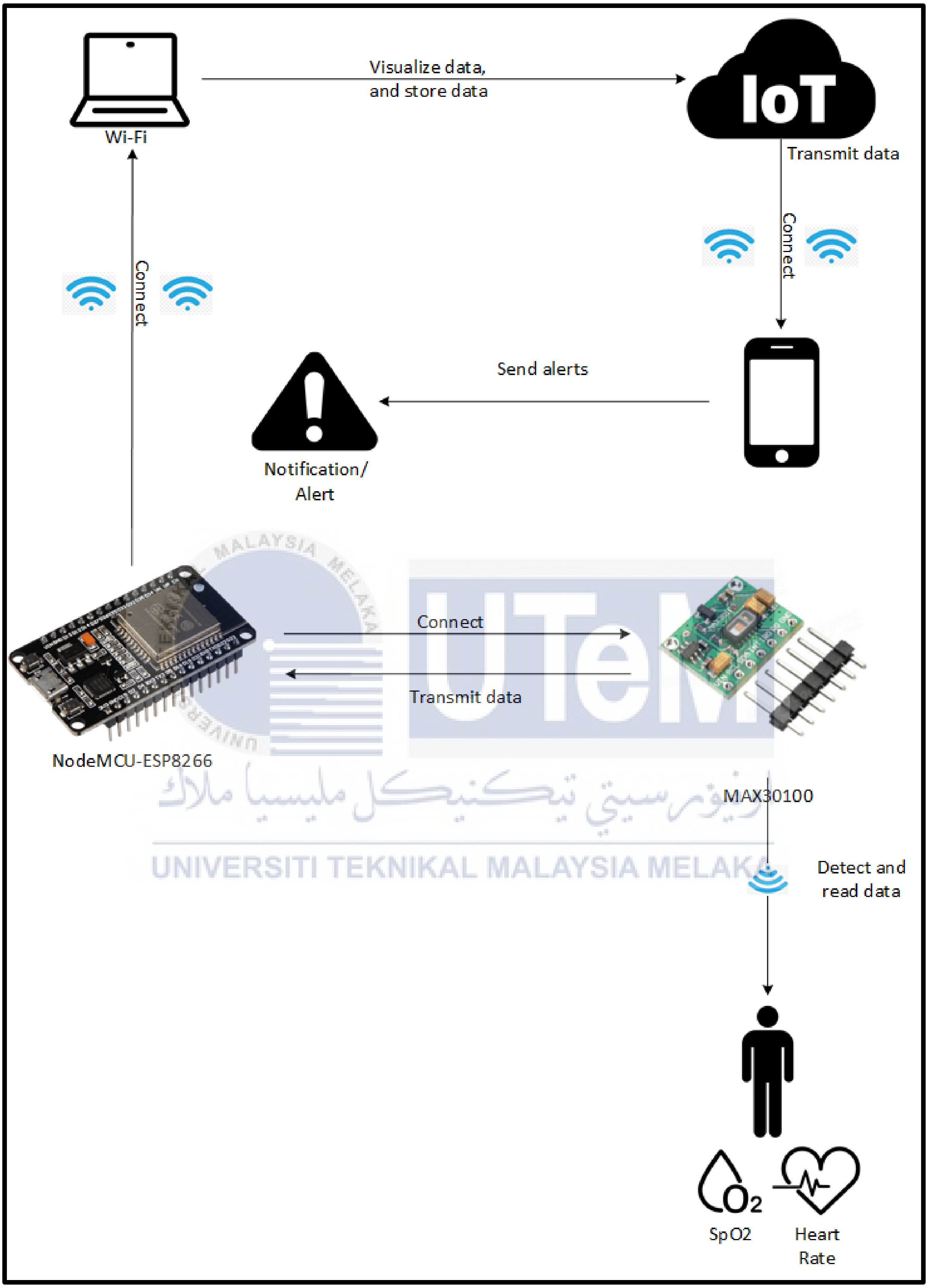


Figure 2.19: Architecture Design for Proposed Solution

2.5 Conclusion

This chapter discusses the project's literature review, which is an existing project review that must be completed to finish a good project. The system development analysis will be explored in depth. Data collection and information, as well as issue analysis and needs, are all included in the literature study. Every outcome is the result of analyzing the data obtained, and some inferences and suggestions have been drawn from this chapter's findings. We may create a comparison from this chapter to provide a better option for the project as a reference.

CHAPTER 3: METHODOLOGY

3.1 Introduction

This chapter went through in detail about the methodology used as a guide and reference to make sure the project is in track and in the correct order. The methodology offers steps and procedures in achieving the milestones while in the project completion process. There are various types of methodologies that can be selected for use as a reference, using the right methodology can determine the overall success and quality of the project report.

Based on this IoT project, there are many types of methodology to be applied such as Agile, Rapid Application Development, Waterfall and Object-Oriented Methodology. In this project, Waterfall model would be used in development of this project. Project milestone is a timeline of a project to be done in making sure that project achieved each progress within the allocated time. Gantt chart will be used to develop planning of the project.

3.2 Methodology

The first thing needs to do on this project is to gain all the information regarding oximeter detection system from journal, articles, books, and internet to provide the best solution for monitoring people health based on their heart rate, and oxygen saturation to detect the symptom of Covid-19 and avoid system respiratory. Collecting data is essential in providing producing a better solution and accurate outcome for this project in both hardware selection and software used. Testing of implemented design into real hardware and software is the most significant phase of this project. It determines whether the project is a success or not success.

The methodology chosen for this project is Waterfall Model which one part of Software Development Life Cycle (SDLC). Waterfall Model phase must be completed before going to the next phase to prevent any overlapping of each phase. Waterfall Model consist of six crucial phase which are requirement analysis, system design, implementation, testing, deployment, and maintenance. The main reason to use Waterfall Model is it determines the goal of the project early. The project scope stays relatively static, meaning cost and timelines can be determined early in the project. Small projects should be very efficient when using this model. Waterfall model also has a clear structure which requires an entire step to be completed before moving to another step.

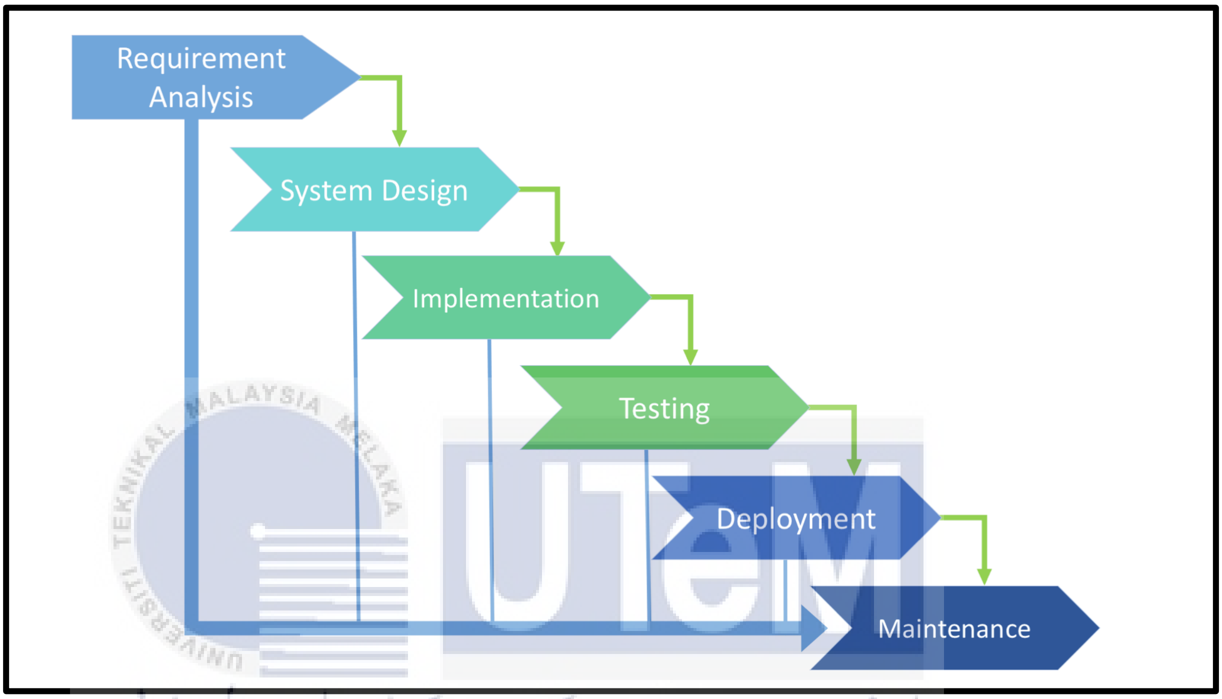


Figure 3.1: Waterfall Model (EXISTEK, 2017)

3.2.1 Requirement Analysis

Each project development life cycle requires to discuss the requirements needed to develop a product. The main objective of this phase is to identify and define the system requirements in detail. All the requirements will be listed and detailed. It is necessary in ensure that process representatives have a clear understanding of the role and how each requirement should be implemented.

Table 3.1: System Requirement

Function	Description
Collect and Monitor Data	The system should be able to collect data from the sensors, and enable user to monitor data collection on real time
Alert Notifications	The system offers alert the user when exceeding the normal values via notification.
Testing and Analysis	The system should be effective for consumerism.

Table 3.2: Hardware Requirement

Item	Description
NodeMCU	Microcontroller kit to implement the coding for communication with IoT devices such as sensors, laptop, and smartphone
MAX30100	Used to measure heart rate, and oxygen saturation value in human body.
Breadboard	Platforms that allow to build and test electronic circuits without having to solder anything.
Jumper Wire	Used to making connections between items on the breadboard and NodeMCU-ESP8266 pins.
Resistor	Reduce the current flow.

Table 3.3 Software Requirement

Item	Description
Arduino IDE	Open-source application compile and run code on microcontroller.
Blynk	Full suite of software required to prototype, deploy, and remotely manage connected electronic devices.
Node.js	Node.js is designed using Google's JavaScript V8 engine for this functionality. Additionally, Node.js also has its own server package, so you do not need to utilize programs like Nginx and Apache. With its event-driven, non-blocking I/O architecture, Node.js can manage many processes at the same time as a thread-based networking platform.

3.2.2 System Design

System design are a phase that discuss about the architecture of the project that helps in defining the overall system using the requirement specify in the previous phase which is requirement phase. The prototype design is to use the sensors to measure heart rate, and oxygen saturation for send data to the NodeMCU-ESP8266. The program code will be written specifically for each sensor. Then, the connection circuit design between the sensor, microprocessor and microchip on the breadboard is drafted.

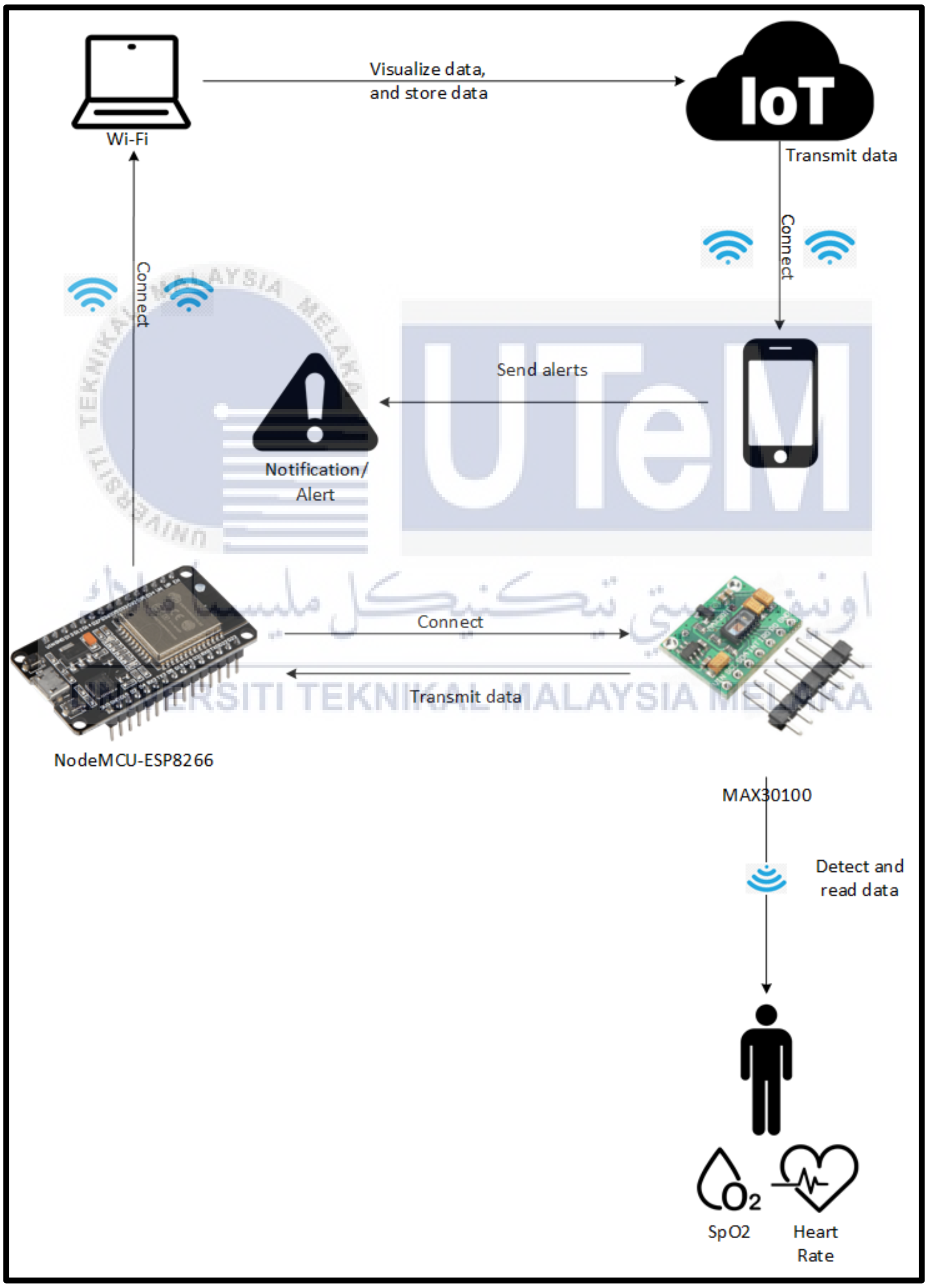


Figure 3.2: System Design

3.3.3 Implementation

The implementation process begins with the connection of each hardware device. Once the connection is complete, the Arduino IDE is installed on the laptop to write the code. All the codes be written to be implement the codes on NodeMCU-ESP8266 by using Arduino IDE which control C++ language development environment. The connection between the sensor and microcontroller is then tested to obtain an output reading.

Once an accurate read is achieved, it then needs to be transferred to the global cloud network in real time. Wi-Fi connection been established on NodeMCU-ESP8266 to ensure the connection to the internet is success. Blynk application will be setting up to receive and sending the data from the NodeMCU-ESP8266. After that, Blynk application will be installed on the smartphone. The application should receive all the upcoming data gain from the cloud Blynk and display it on the smartphone.

3.3.4 Testing

In this phase, the testing process been conducted to ensure that everything goes in line with the expected results and goals. To test the code, the sensor, and microcontroller will be connected to determine accuracy of the sensor. Blynk application will be installed in a smartphone and need to configuration to be done and enable connection to the microcontroller. The application should receive all the upcoming data gain from the cloud Blynk and display it on the smartphone.

3.3.5 Deployment

Once the functional and non-functional of the hardware and system had been tested, deployment of this project in real environment is ready to be tested. This phase is done once the testing has no bug or error or any issues.

3.3.6 Maintenance

The main objective of this phase is to maintain the system and hardware are functional. Monitoring this product are compulsory by the author to fix any bugs and errors sent by the end user. It is essential to give new update of the product when it is available to the end user. Any feedback and review will be sought to improve and correct any issue from the developed application.

3.4 Project Milestones

Table 3.4: Project Milestone

Phase	Activity
Phase 1: Requirements	• Project Proposal • Proposal assessment & verification • Proposal improvement and correction • Proposal submission • Identify the objectives, scope and problem statement. • Project requirement identification
Phase 2: System Design	• Design the architecture of the project. • Define the connection to be made in the projects. • Determine the system environment needed for the device
Phase 3: Implementation	• Write and upload code into the device. • Build hardware connection. • Set up connection with Blynk
Phase 4: Testing	• Monitor the system for any error and flaw.

	• System is tested into real environment scenarios
Phase 5: Deployment	• Test the prototype with users. • Collect feedback from users
Phase 6: Maintenance	• Monitor the product for any errors and bug

Table 3.5: Project Gant Chart

Task Name	Week																				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Requirements Analysis																					
System Design																					
Implementation																					
Testing																					
Deployment																					
Maintenance																					

3.5 Conclusion

To summarize, methodologies may be described as structured of steps, techniques, design project, and processes, components, and perspective. Different project has different techniques to solve the problem. By taking the main used in my project that was ESP8266, and MAX30100 sensor then I can configure the program without need to purchase or subscribe the premium application or code. This is because the platform has intentionally created in open-source system to attract researcher or inventor to invent creative product. Next, Android application is free, easy-to-use to configure alert to the user. Even though, there are some lacks feature in this project, and there are positive issues we should look at and negative issues that need to be countered.

CHAPTER 4: ANALYSIS AND DESIGN

4.1 Introduction

The result of the analysis of the preliminary design and the outcome of the detailed design will be described in this section. It will also concentrate on the study of project specifications. The block diagram for this project will also be presented to ensure that the plan can be completed and well planned. Specifications such as the hardware and software required for this project are included in this chapter.

4.2 Problem Analysis

Nowadays, we have a wearable smart device oximeter that can monitor heart rate and oxygen saturation that can be used as fitness devices and measures to reduce respiratory system like Happy Hypoxia due to Covid-19, but because some people may not be able to buy the commercial oximeter and smartwatch that was built up to check for oxygen saturation in people's bodies. People may also be uncertain whether their body is healthy or not since the oximeter sometimes just displays the oxygen saturation data but does not provide the warning when their SpO2 is not in excellent condition. The present oximeter may not be able to store data since it simply does not contain the software.

Current Pulse Oximeter System

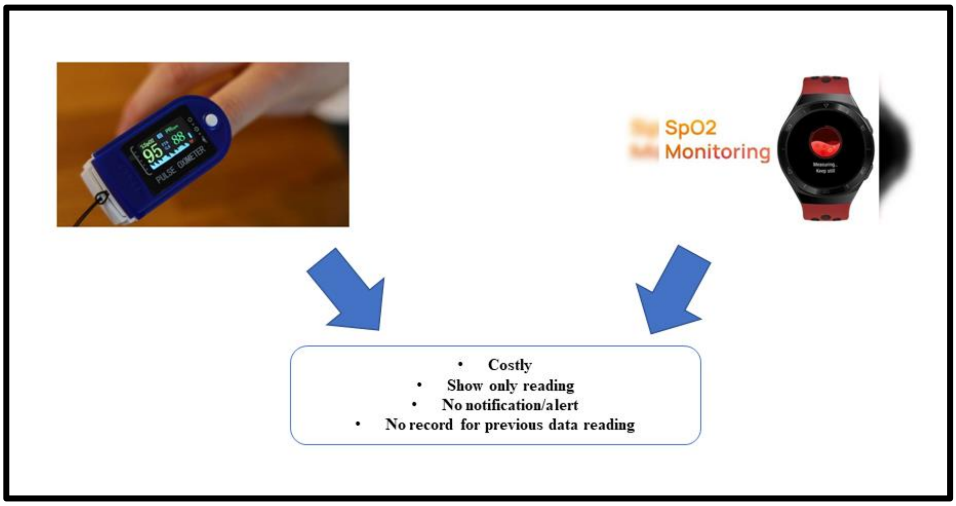


Figure 4.1: Current Pulse Oximeter System

Current devices on the market are costly for a customer to purchase. Because this device lacks IoT technology that can alert and record previous data, it only displays data readings.

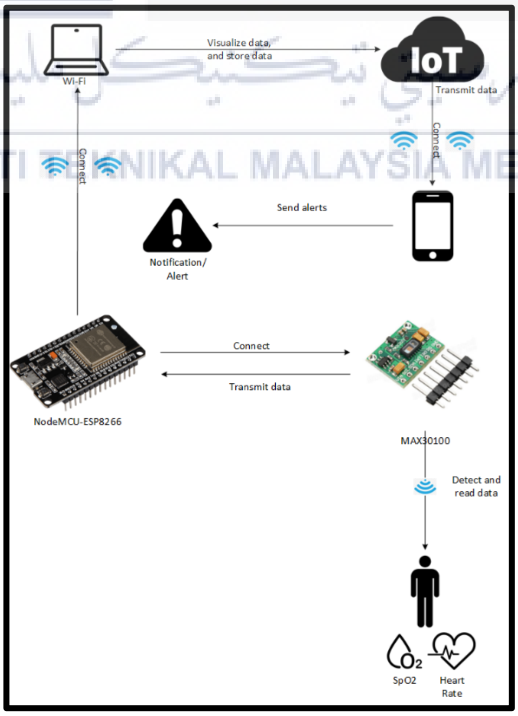


Figure 4.2: Proposed Pulse Oximeter System

We developed a system named Development of DIY Oximeter Health Monitoring Tools utilizing heart rate and oxygen saturation sensor. This device is super-cost efficient and allows users to monitor heart rate and oxygen saturation. This device also provided the user with an alarm for an emergency. This project helps users monitor their elderly and assisted the old notify their near contact.

4.3 Requirement Analysis

4.3.1 Data Requirement

The oxygen saturation and heart rate MAX30100 are the system's inputs. The MAX30100 sensor measures the proportion of SpO2 and BPM in the human body. Through the NodeMCU microcontroller, data will be detected, transmitted, and recorded in the Blynk Cloud. This system's output is an alarm to the user when oxygen saturation and heart rate levels are exceeded. As the future, the data will be generated in a csv file for the user to see. The SpO2 and BPM were shown in real time using a Blynk smartphone application.

Figure 4.3 below show the project data requirement, Blynk is a platform with IOS and Android application to control IoT microcontroller board with access of the Internet. It is a digital dashboard where it can build a graphic interface for the user by simply drag and drop the widgets. Blynk application react as bridge between the hardware and the Blynk server. The data of the hardware the send to Blynk apps automatically be stored in the Blynk IoT cloud. The help from Super Chart widget it also can be export it as csv type for the user to read it. In this project, default Blynk server is used to store all the data from the sensor. So, user can track the data in graph display in Blynk application. Figure 4.3 below show the project data requirement.

Figure 4.3: Project Data Requirement

4.3.2 Functional Requirement

- **User Login**

  Description: The system will provide the widgets, and profile page to design their system.

  Input: Enter Email and Password

  Output: User Profile Page

  Processing: The system will check the input of the user and if it valid, then it will login but if it not valid, the user needs to renter again email and password.

- **Sign Up**

  Description: The system will the access for login and provide user account.

  Input: Enter Email and Password

  Output: User Account

  Processing: The system will check the email if available or not for sign up. If the email available user to user, the other email.

The smartphone will receive alert notification by the NodeMCU-ESP8266 which connected to the breadboard and sensor. The MAX30100 sensor will be used to measure the heart rate, and oxygen saturation. The function of the smartphone is to display the data from the sensor and to be used for alert notification.

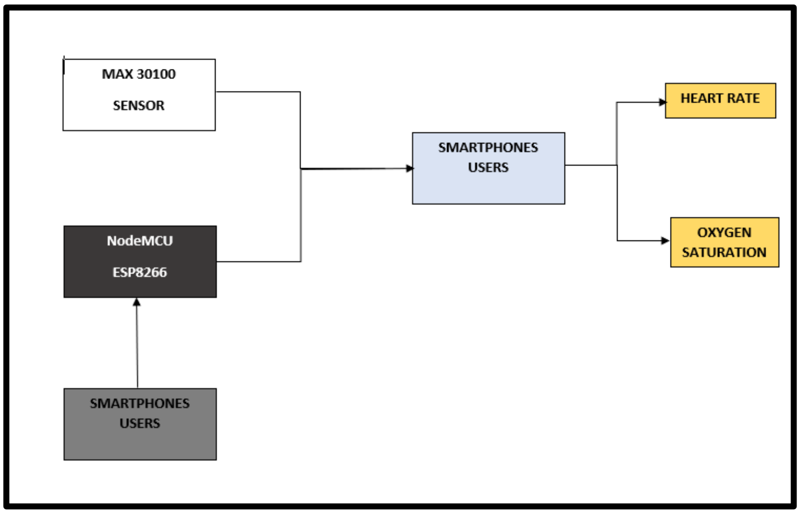


Figure 4.4: Functional Diagram

4.3.3 Non-functional Requirement

This project contains a few quality requirements that is reliability, consistency, availability, and usability.

- **Reliability**

  The MAX30100 sensor module, the exact heart rate and oxygen saturation will be detecting as 90% accurate and the Blynk app will shows all the sensor data.

- **Consistency**

  If the user using the hardware the data will display consistently, and the data will automatically be stored in the Blynk cloud.

- **Availability**

  Blynk application service will be up to 99.99% of the time.

- **Usability**

  Blynk application is easy to be used and to be understood, the interface contains the display of body temperature data and pulse rate data. Graph of both data also included for the user interface. The data can be easily export through the Blynk server.

➢ **Security Requirements**

The user profile can be access if the email and password is correct because the data must be safe and secure from alter.

4.3.4 Others Requirement

Software Requirement:

➢ **Arduino IDE**

Arduino IDE is an open-source software which is mostly used for writing and compiling code into the Arduino module. It is easily available for operating systems such as MAC, Windows, Linux, and runs on the Java Platform which comes with built-in functions and commands that play a vital role in debugging, editing, and compiling code in the environment. This environment supports both C and C++ languages.

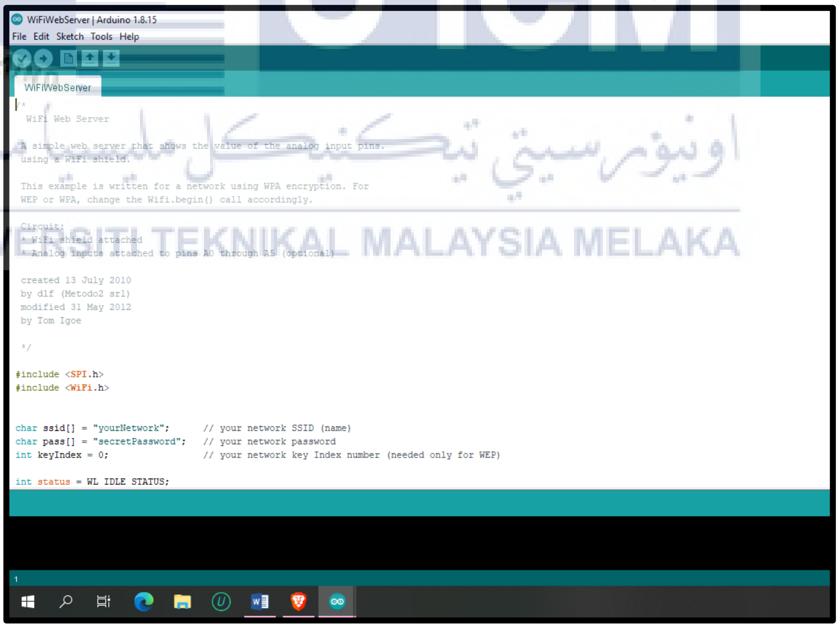


Figure 4.5: Arduino IDE

- **Blynk Application**

Blynk is a platform for managing NodeMCU, Arduino, Raspberry Pi, and the likes of IOS and Android devices via the Web. You can create an interactive framework for your project by simply drag and drop widgets. This is visual dashboard. In this project, I will use Blynk application to monitor and stored the data from the sensor. Blynk support the following connection types to connect your microcontroller board (hardware) with the Blynk Cloud, and Blynk personal server.

Figure 4.6: Blynk Application

- **Node.js**

Node.js is Ryan Dahl's framework to execute JavaScript-based web applications, launched in 2009. With this platform, you may execute server-side JavaScript. Node.js is designed using Google's JavaScript V8 engine for this functionality. Additionally, Node.js also has its own server package, so you do not need to utilize programs like Nginx and Apache. With its event-driven, non-blocking I/O architecture, Node.js can manage many processes at the same time as a thread-based networking platform.

Figure 4.7: Node.js

Hardware Specifications:

- **NodeMCU**

The NodeMCU is one of the low-cost IoT platform which runs with ESP8266 Wi-Fi module. It is also the most robust and easy to use the boards since it is small.

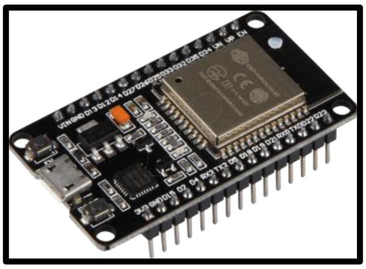


Figure 4.8: NodeMCU (ESP8266)

- **Jumper Wire**

Jumper wires are simply wiring with connector pins at each end. Jumper wires allowing to be used to connect two points to each other without soldering, and generally used with breadboards, solderless board, and devices to make it easy to modify circuits when required. Jumper wires typically come with three versions that is male-to-male, male-to-female, and female-to-female.

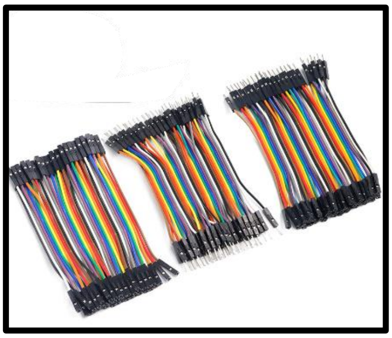


Figure 4.9: Jumper Wire

➢ **Heart Rate and Oxygen Saturation Sensors**

MAX30100 is a heart rate monitor. Software registers may configure this module and store its output data on this module in seventeen FIFOs. This device is used to monitor heart pulse rate and blood oxygen concentration, since these measures wear this sensor on your finger or earlobe, it may also place any area of the body that is not thick. Your finger is measuring both infrared and red light via a finger's tissues, these lights may be measured by a photodiode. The ratio of absorbed red light and infrared light will vary depending on the amount of oxygen in your blood and can simply compute the oxygen level in your blood hemoglobin.

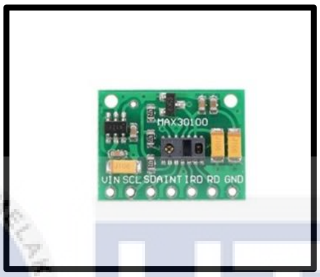


Figure 4.10: MAX30100 Sensor

➢ **Resistor**

Resistor used to reduce the current flows. High power resistor that can dissipate many watts of electrical power.

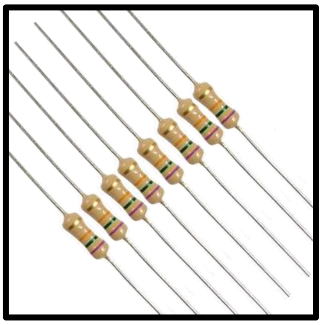


Figure 4.11: Resistor

➢ Breadboard

Breadboard is the bread-and-butter of DIY electronics. Breadboards allow beginners to get acquainted with circuits without the need for soldering, and even seasoned tinkerers use breadboards as starting points for large-scale projects.

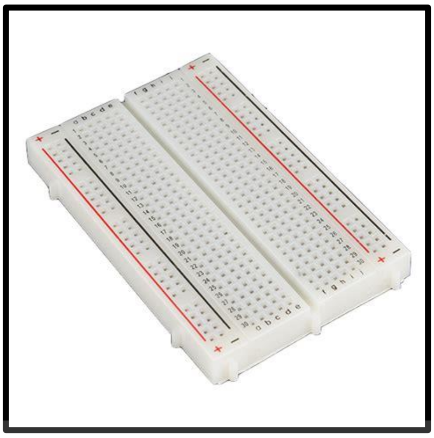


Figure 4.12: Breadboard

➢ OLED Display

The OLED (Organic Light-Emitting Diode) display is an alternative for LCD display. The OLED is super-light, almost paper-thin, flexible, and produce a brighter and crisper picture. There are many types of OLED display. They differ from each other in communication interface, sizes, and colors:

a) Communication interface: 12C, SPI
b) Size: 128x64, 128x32
c) Color: white, blue, dual color

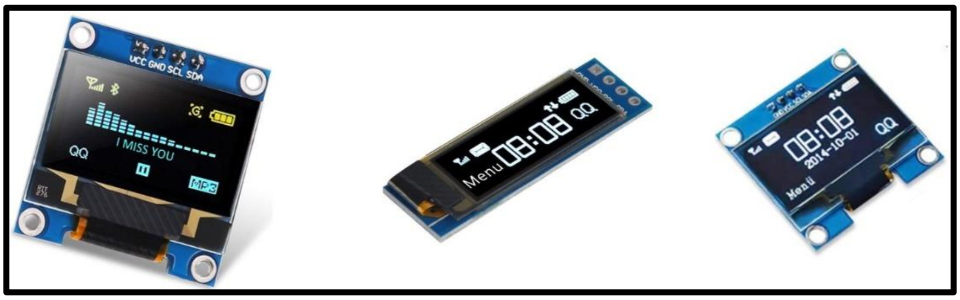


Figure 4.13: OLED Display

4.4 High Level Design

Development DIY Oximeter for Health Monitoring system interacts with its environment using physical sensors. This requires a bridge between the physical sensor and the software layer. The firmware is the software responsible for interaction with the real world. A NodeMCU-ESP8266 based on microcontroller runs the firmware and execute sensors.

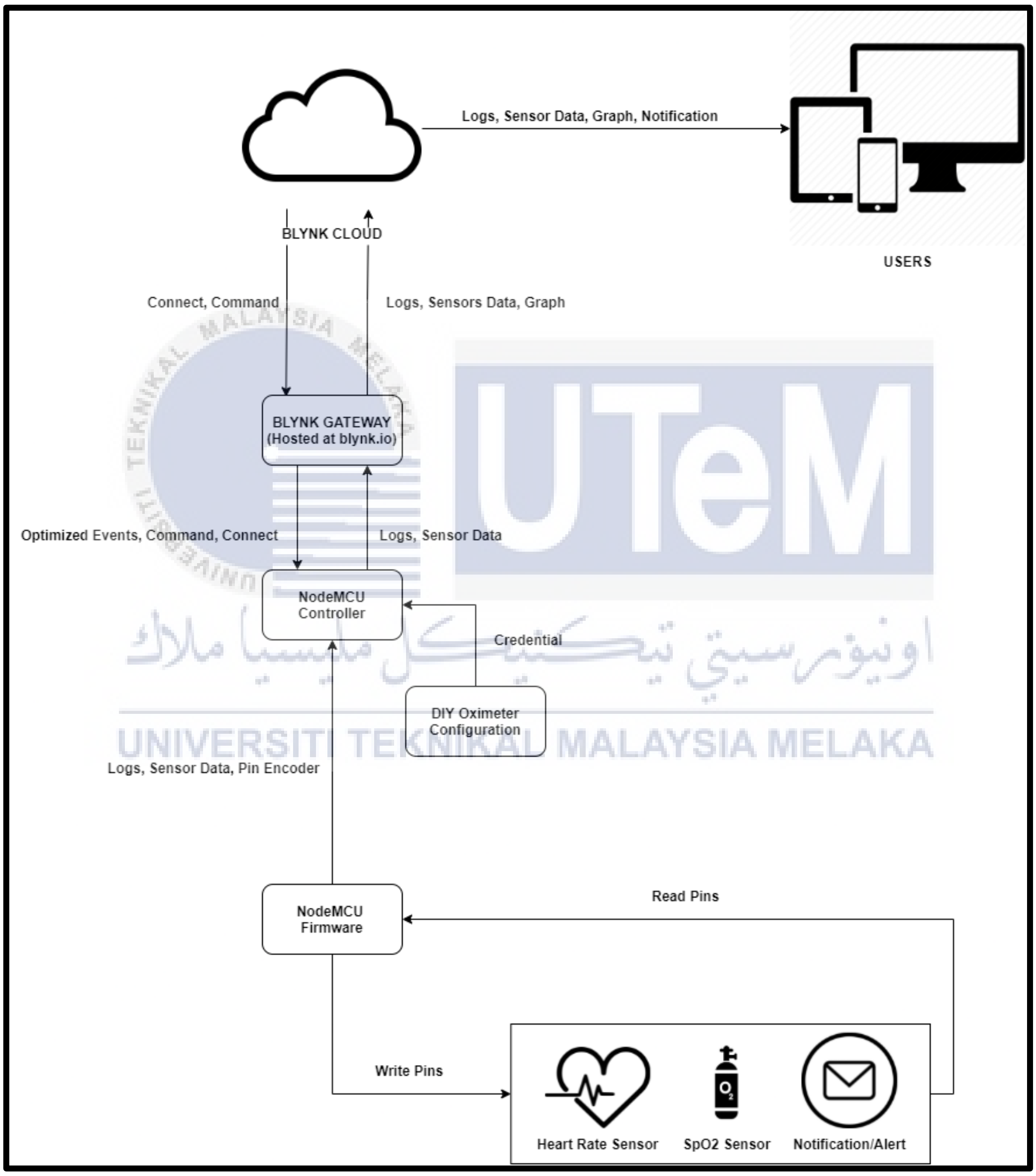


Figure 4.14: High Level Design Diagram

4.4.1 System Architecture

This system uses NodeMCU as microcontroller which will programmed to read the MAX 30100 (heart rate and oxygen saturation) sensor to obtain the heart rate and oxygen saturation of the users. The data will be sent to the user online through ESP8266 Wi-Fi module to the smartphone. This system also features a notification alert on a designated condition of the heart rate if the threshold value is surpassed using Blynk application. This system will use a micro-USB as power supply.

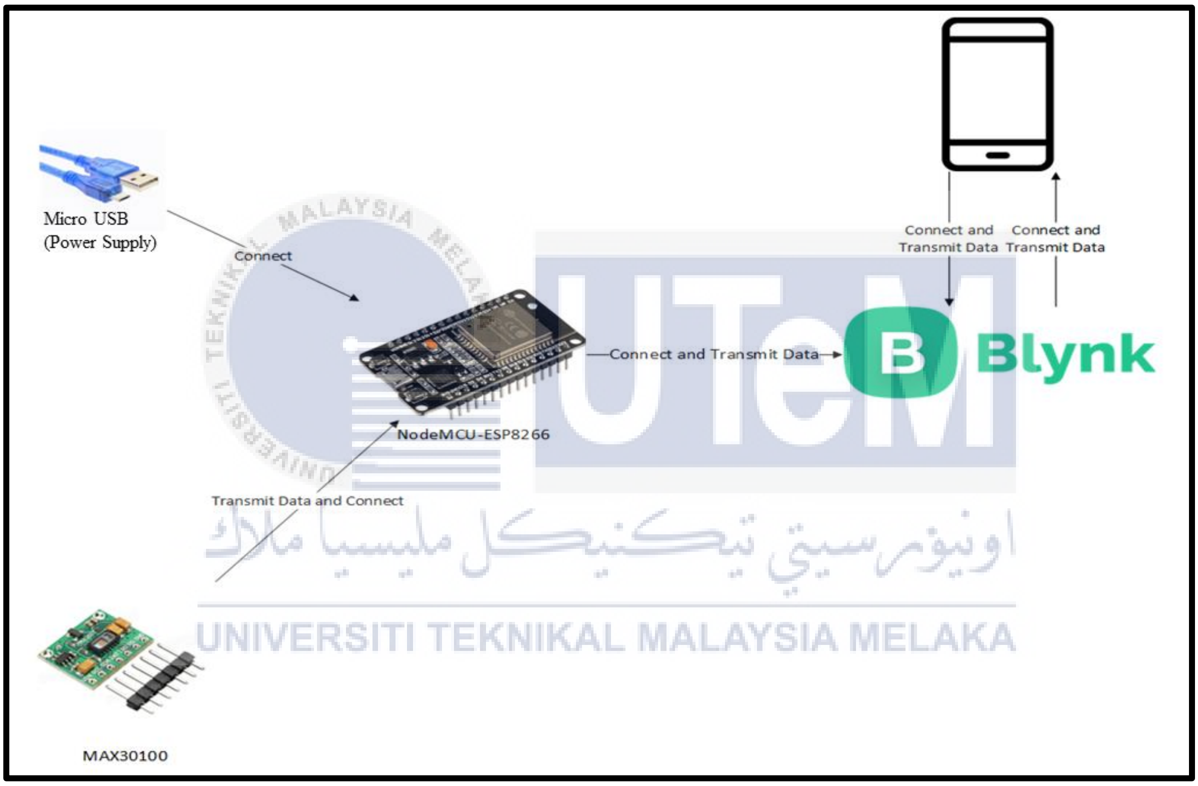


Figure 4.15: System Architecture

4.4.2 User Interface Design

a. Navigation

i. The user will view the login screen before entering an email and password. The default user will be allowed to access the MAX 30100 sensor to measure heart rate and oxygen saturation.

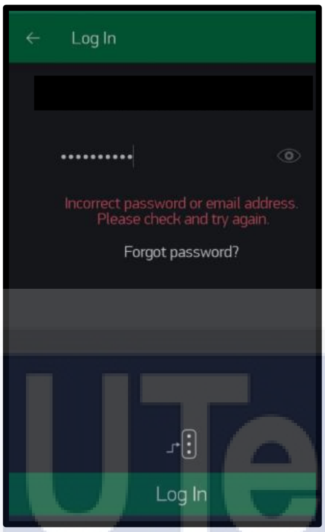


Figure 4.16: Login Interface of Development of DIY Oximeter for Health Monitoring

ii. After logging in, the complete Development of DIY Oximeter for Health Monitoring interface appears. Then, the user must click the play button to link the software and hardware together.

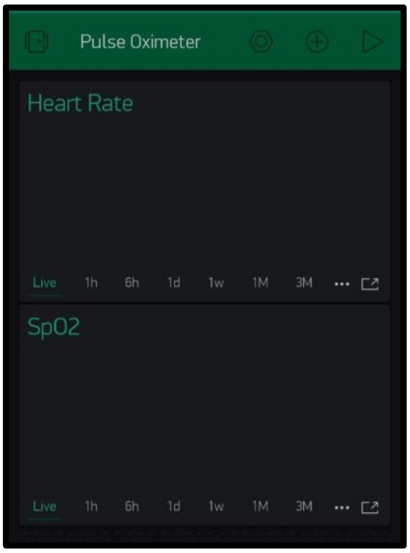


Figure 4.17: Login Interface of Development of DIY Oximeter for Health Monitoring

iii. It will maintain updated live statistics of heart rate and oxygen saturation after clicking the play button.

Figure 4.18: Monitoring Interface Deveploment of DIY Oximeter for Health Monitoring

iv. The user may choose to export the heart rate and oxygen saturation data to an email or delete the data history.

Figure 4.19: Exporting the Graph of Data Monitoring

b. Input Design

i. Heart Rate (BPM) widget

This input display widget enables Arduino to send Blynk heart rate data. The source code uses the same virtual pins for this input, V0.

Figure 4.20: Input Display Widget of Heart Rate Reading

ii. **SuperChart Widget**

This widget reads NodeMCU heart rate and oxygen saturation. Using timers, the hardware data will be pushed at the appropriate interval. This widget may display current graphs and historical statistics. This widget has two virtual pins: V1 for heart rate and V2 for oxygen saturation.

Figure 4.21: Input Display Widget of SuperChart

c. Output Design

i. This is the history graph of the MAX 30100 sensor's heart rate and oxygen saturation. An hourly, daily, weekly, monthly and 3-month graph may be shown using SuperChart widget.

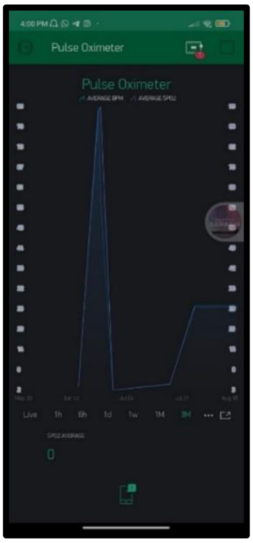


Figure 4.22: Historical Graph

ii. After the user clicks the export data button, an email is immediately sent from the Blynk server to the email address associated with the user's Blynk account.

Figure 4.23: SuperChart Data

4.5 Detailed Design

This section shows the detailed design of the project where it will contain the structure view of the system to be developed which including the circuit diagram, flow chart of the system, and pseudocode.

4.5.1 Circuit Diagram

Prototyping is a big part in project, and it is a thing that very difficult. Circuit design can keep going over things and fixing the connection mistakes. Circuit design is a very important for a good project prototype. Here is the sketch of Development DIY Oximeter for Health Monitoring circuit design using online software before the real hardware was applied.

For this connectivity, the NodeMCU will use the port 3V3, GND, D2, D1, D0, OLED display will use the port SCL, SDA, VCC, GND, and the MAX30100 sensor will use port VIN, SCL, SDA, INT, GND. The port GND from NodeMCU will be place on the GND port MAX30100 sensor, and OLED display GND port. The port 3V3 from NodeMCU will be place on the VIN port MAX30100 sensor, and OLED display VCC port. The port D0 from NodeMCU will be place on the INT port MAX30100 sensor. The port D2 from NodeMCU will be place on the SDA port MAX30100 sensor, and OLED display SDA port. Finally, the port D21 from NodeMCU will be place on the SCL port MAX30100 sensor, and OLED display SCL port. The figure 4.19, 4.20, 4.21, 4.22 show the detailed of connectivity for this circuit.

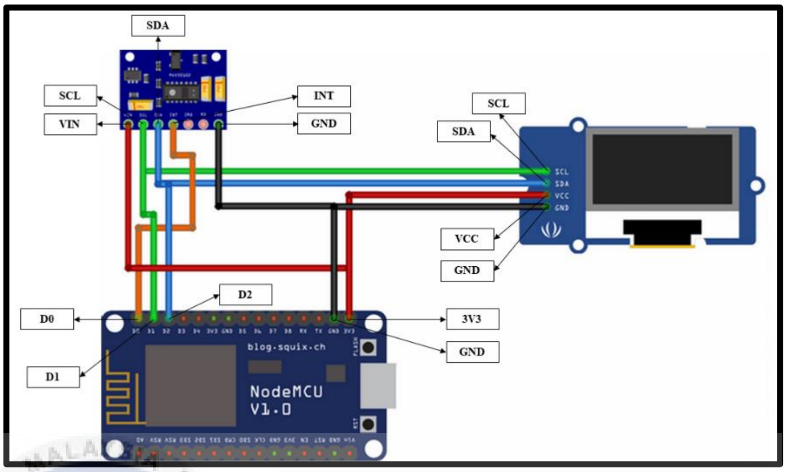


Figure 4.24: Circuit Diagram

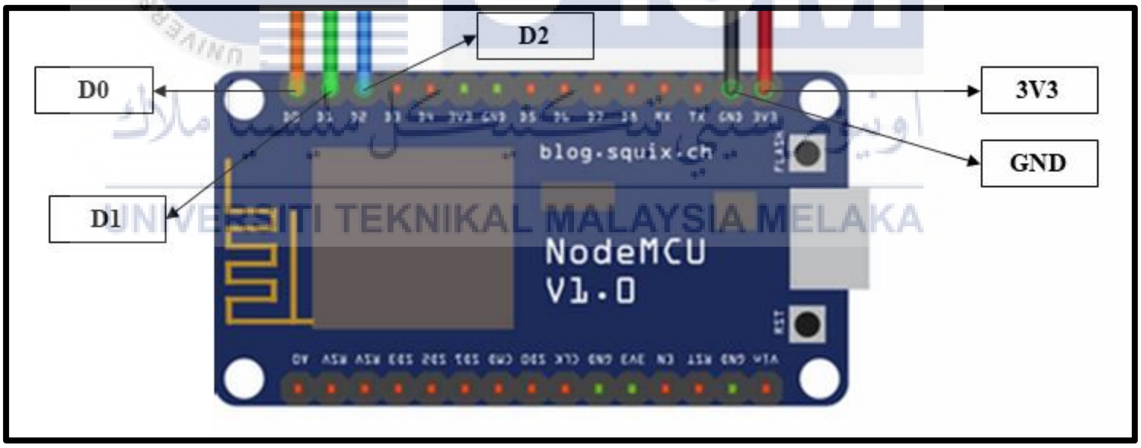


Figure 4.25: Circuit Diagram for NodeMCU

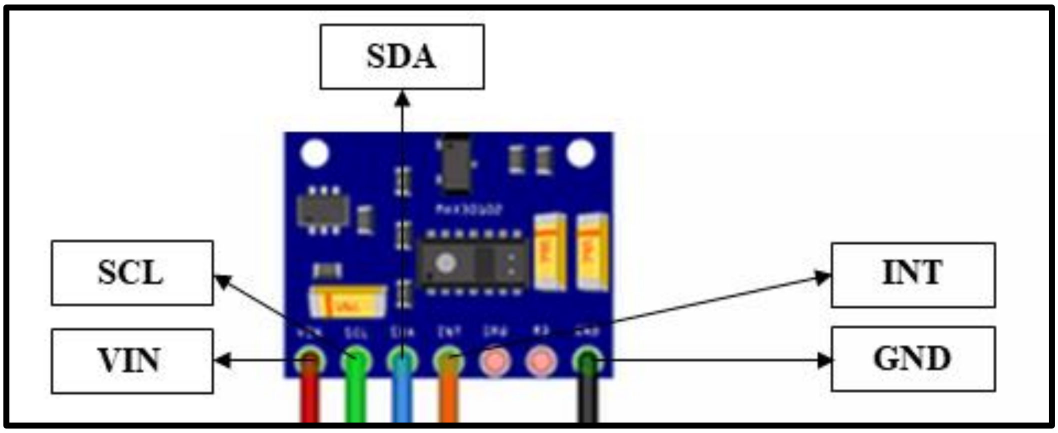


Figure 4.26: MAX30100 Circuit Diagram

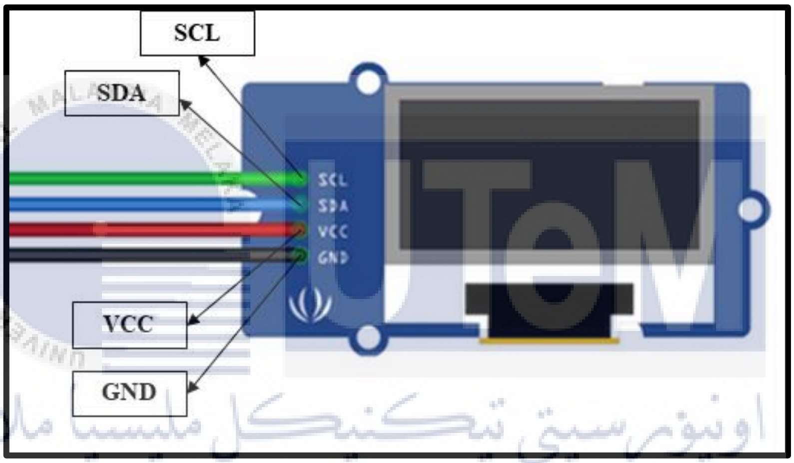


Figure 4.27: OLED Display Circuit Diagram

4.5.2 Flow Chart

Flowchart is a formalized graphic representation of a logic sequence of the system. In the design phase flowchart is one part of the project development. This flowchart shows the flow of the system functionality start from the device to the monitoring part of the project.

Firstly, before utilizing this project, we must first activate this project using micro-USB as a power source. We must then place a finger on the sensor to measure oxygen saturation and heart rate. OLED display, and Blynk will display data from the sensor to be projected within the phone. Here we have 2 criteria if SpO2 is low from 95 percent and BPM is high from 140 percent, the user will be warned to act and if our SpO2 and BPM are still in normal and show to the Blynk application on smartphone. Here we have 2 criteria saving data previous data if user say yes, it will be sent to email and save in csv file, and if user say no it will back display data on Blynk.

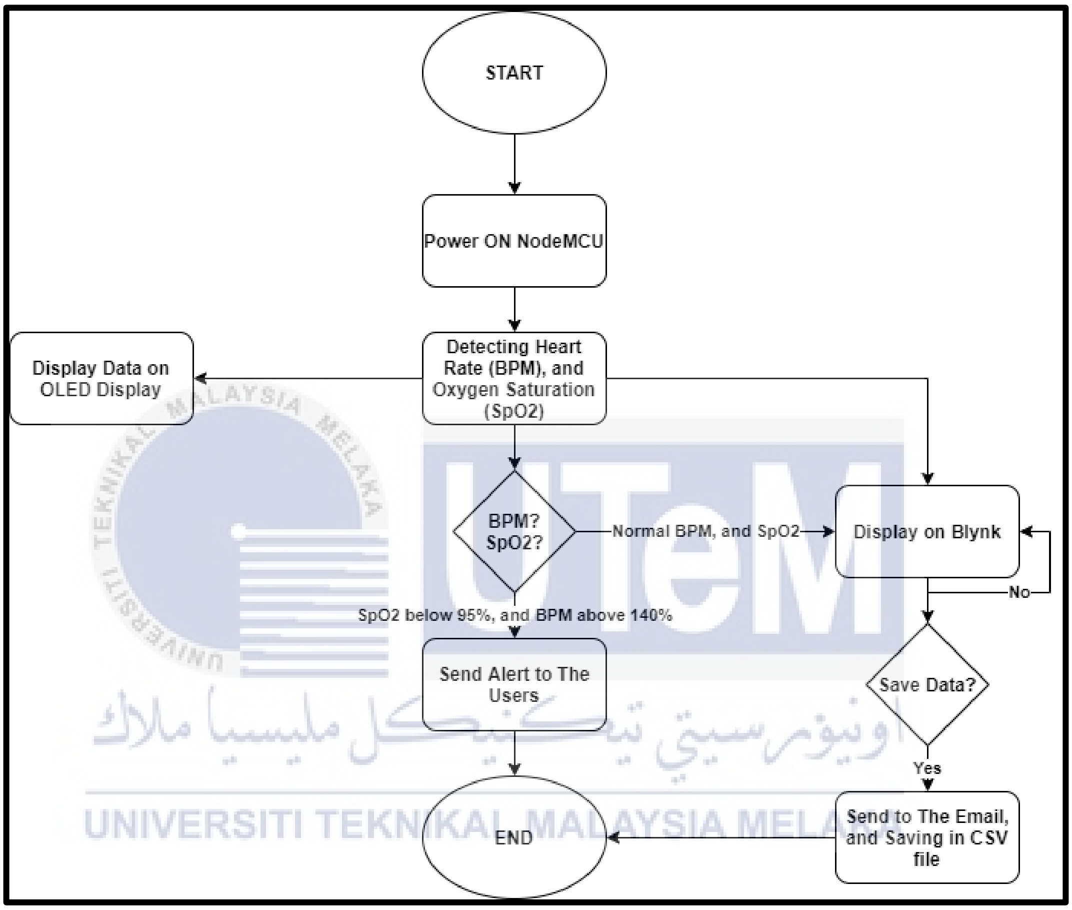


Figure 4.28: Flowchart System

4.5.3 Pseudocode

Pseudocode is a basically description of algorithm or codes in a plain language. It is intended for human to understand the code rather than machine. Pseudocode is understandable compared to any codes that uses algorithm.

This pseudocode program must be aware of the project's input and output, since it is critical to understand what the sensor reads. The project's inputs are pulse and oxygen saturation, as we want to collect data. For the output, it is possible to utilize a variety of variables, including BPM and SpO2, since these two values will be shown in the Blynk program. Following that, the program may be launched since the pseudocode's inputs have been identified, namely oxygen saturation and pulse, which will be read by the sensor. To generate output, we will transform the result of the input into bpm for heart rate and SpO2 for oxygen saturation. Finally, utilizing inputs that have been transformed to outputs, utilize the if condition to generate the two most recent outputs, namely alerts and data records.

```
Program to detect User's Health
Initialize
          Input
             Pulse
             Oxygen saturation
          Output
             Heart rate
             SpO2

Start Loop
Detect pulse and oxygen saturation
Convert to heart rate in bpm, and SpO2 in oxygen
          if bpm > 140
                    send alert to the user
          if oxygen < 95
                    send alert to the user
          else
                    display heart rate, and SpO2
end loop
```

Figure 4.29: Pseudocode System

4.5.4 Deployment View of DIY Pulse Oximeter System

The deployment perspective of the system is shown in Figure 4.21, which demonstrates that the DIY Pulse Oximeter System was built utilizing the Blynk program, which is accessible to all users. The user will be linked to the Blynk application to access the data, and the data will be saved in the cloud as a csv file.

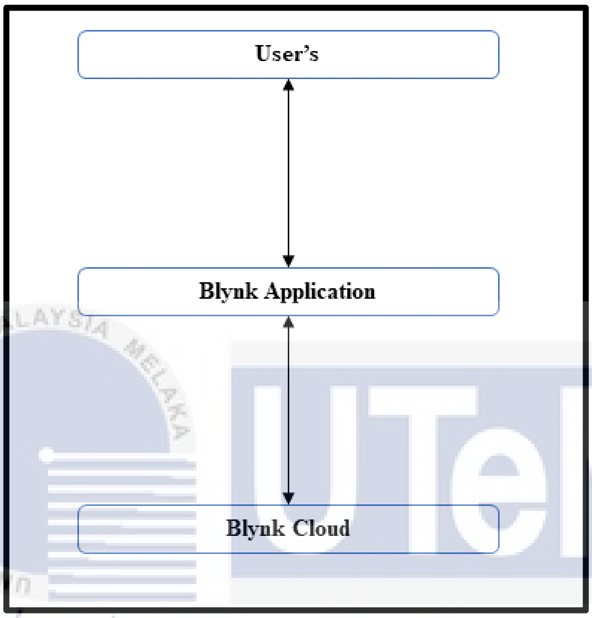


Figure 4.30: Deployment View of DIY Pulse Oximeter System

4.6 Conclusion

Analysis and design are crucial parts of the development of the project. All software and hardware specifications must be defined and examined before the implementation of the design. This chapter is the pre-preparation phase for application which provides the flow of the entire system so that it can be better understood before it is implemented. This chapter also includes all the important design requirements as the part that needs to be implementing and testing in the next chapter. It also explains the design of the architecture for this project that has been planned.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

This chapter will go over how to use IoT to develop a DIY Oximeter for Health Monitoring with notification and data storage through email in both software and hardware development. The microcontroller in this project is a Node-MCU, and the sensor is a MAX 30100 that detects bodily oxygen saturation (SpO2) and heart rate (BPM). Finally, to make this project function, it incorporates additional software and hardware into the IoT to develop a DIY Oximeter for Health Monitoring System.

5.2 Software Development Environment Setup

This section will discuss the project's surroundings briefly. This project needs a prototype of the DIY Oximeter for Health Monitoring system, which includes all necessary hardware and software. The prototype will detect the oxygen saturation (SpO2) and heart rate (BPM) and then show the data graphically and save it to a CSV file.

5.2.1 Prototype Development

In the prototype development, the Arduino hardware used is NodeMCU with OLED display, MAX 30100 sensor, jumper wires, and resistor.

i. NodeMCU

NodeMCU is the core platform for this project because its act as the center of all process. The data from the sensor sends to NodeMCU, then the data is calculated and will be sent by NodeMCU to the smartphone using Wi-Fi and display on the Blynk application system. The NodeMCU connect to a breadboard and will be using micro-USB to connect NodeMCU and laptop.

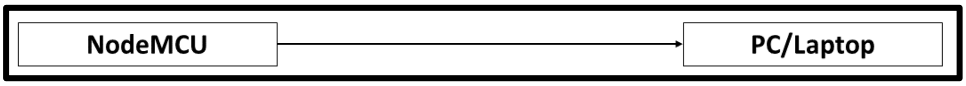


Figure 5.1: Block Diagram for NodeMCU Environment

ii. **MAX 30100 Sensor**

MAX 30100 sensor is a sensor to measure heart rate and oxygen saturation. Sensor connected to the breadboard and the jumper wire is used to connect the sensor and NodeMCU to the breadboard. The breadboard allows the electric current to flow from the power source to the NodeMCU and the sensor.

Figure 5.2: Block for MAX 30100 Sensor Environment

iii. **OLED Display**

The OLED display is an Arduino hardware to display reading from the data they get from the sensor or other hardware and display it on OLED display.

Figure 5.3 Block Diagram for OLED Display Environment

5.2.2 Software Development

In software development, the Blynk application is used to store and get data from NodeMCU to used real-time monitoring for this project. The Blynk application has a Blynk server used for the NodeMCU to connect with this project system to transfer data using a Wi-Fi signal.

i. **Blynk Application**

Blynk was created with the Internet of Things in mind. It can remotely control devices, display sensor data, store, and visualize data, and do various other exciting tasks. The Blynk application lets you build great user interfaces for your projects by using the numerous widgets we offer. In contrast, the Blynk server manages all connections between the smartphone and hardware. You may either utilize our Blynk Cloud or host your private Blynk server on your hardware. It is open source and can support thousands of devices. This program must be downloaded from the Google Play or Apple App Store on a smartphone before installing on an Android or iOS device.

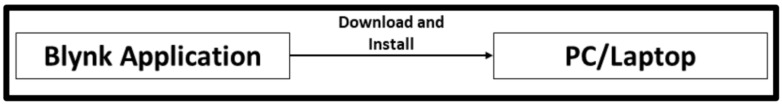


Figure 5.4: Block Diagram for Blynk Application Environment

5.3 Software Configuration Management

The project will be fully described in this section, including the design and implementation of the configuration management system for this project. The discussion will include an overview of the software and hardware tools utilized to assist our configuration control process.

5.3.1 Prototype Configuration Setup

To enable all the Arduino hardware that connects to the NodeMCU to carry out their function of Arduino IDE software need to be installed first before typing the program into the NodeMCU. C programming language is the language which the Arduino is used. Besides, this section also explained the code which is used in the Arduino hardware.

i. Arduino IDE Installation

Detailed Arduino IDE Installation steps as show in below:

1. Download the Arduino IDE platform at https://www.arduino.cc/en/software

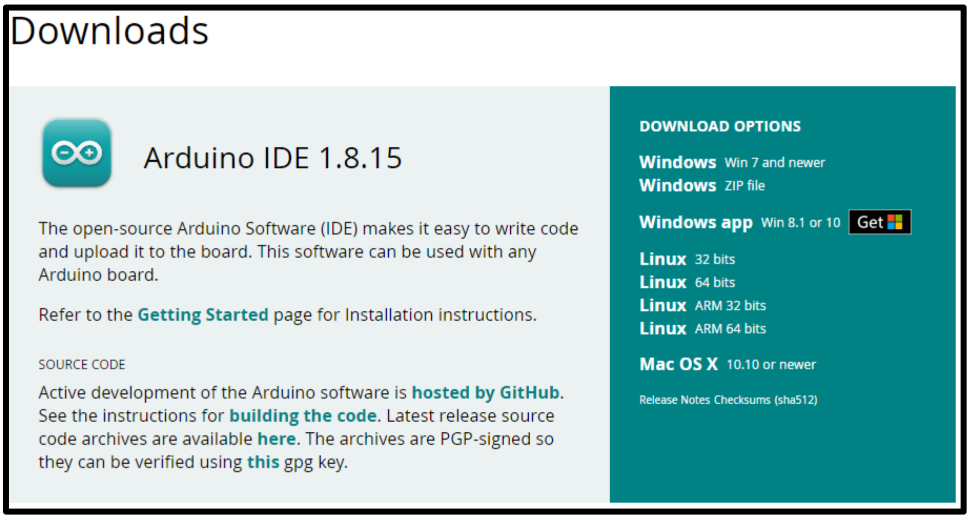


Figure 5.5: Arduino Website Download

2. Extract the downloaded file, open Arduino setup to begin the installation. After that installation, run the Arduino IDE software.

Figure 5.6: Arduino IDE Initialize

3. After a few minutes, it will display a coding interface where the code put and upload into the NodeMCU or Arduino. **C:\Users\PermataHikmah\OneDrive - University Technical Malaysia Melaka\Documents\Arduino** are the location path for code.

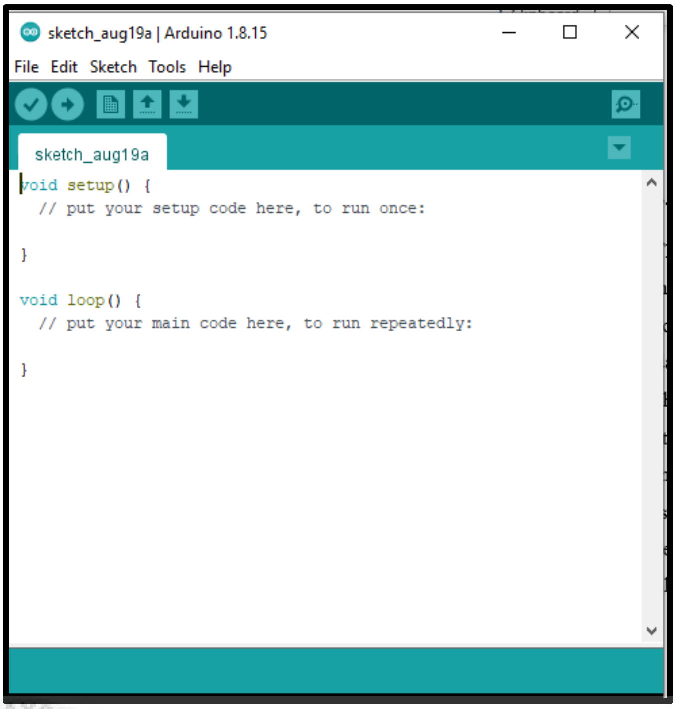


Figure 5.7: Arduino Code Platform Interface

4. Connect NodeMCU to the laptop.
5. Ready to program the NodeMCU.

ii. Prototype Configuration Setup

a. Declare Library

Before coding, you must first declare specific libraries, such as the jumper wires library, the OLED display library, the ESP866 Wi-Fi library, and the Blynk library, as shown in Figure 5.8. This library is critical since it contains the NodeMCU component, sensor functionality, and Blynk; without it, the NodeMCU component, sensor, and Blynk will not work correctly.

```
#include "MAX30100_Filters.h"
#include <Wire.h>
#include "MAX30100_PulseOximeter.h"
#define BLYNK_PRINT Serial
#include <Blynk.h>
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

#include "Wire.h"
#include "Adafruit_GFX.h"
#include "OakOLED.h"
```

Figure 5.8: Declare the Library

b. Declare Wi-Fi SSID and Password

Next, declare Wi-Fi SSID and password to ensure the NodeMCU can connect to the Wi-Fi, as shown in Figure 5.9.

```
char ssid[] = "phikmah_2.4@unifi"; // Name of your WiFi SSID
char pass[] = "ezlot7428"; // WiFi Password
```

Figure 5.9: Declare Wi-Fi SSID and Password

c. Declare Blynk Token

The Blynk token, obtained via the Blynk app on a smartphone, must be declared in the code. The Auth Token is a one-of-a-kind identification that is required to link your gear to your smartphone. Your Auth Token will be unique to each new project you create.

```
char auth[] = "5TK_IaaDBdmhYsnZt_wxKlJdOf4O3uvr"; // Blynk authent
```

Figure 5.10: Declare Token Auth

d. Code for the OLED to Display Reading from MAX 30100 Sensor

The void my timer event, the code in Figure 5.11, is used for the OLED to display the reading from the MAX 30100 sensor. The OLED will print the “SpO2” and “BPM” if the NodeMCU successfully uploads and connect to the Wi-Fi.

```
oid myTimerEvent() {
Blynk.virtualWrite(V7, BPM);
Blynk.virtualWrite(V8, SpO2);
Blynk.virtualWrite(V9, BPMAvg);
Blynk.virtualWrite(V10, SpO2Avg);

 oled.clearDisplay();
 oled.setTextSize(1);
 oled.setTextColor(1);
 oled.setCursor(0,16);
 oled.println(pox.getHeartRate());

 oled.setTextSize(1);
 oled.setTextColor(1);
 oled.setCursor(0, 0);
 oled.println("Heart BPM");

 oled.setTextSize(1);
 oled.setTextColor(1);
 oled.setCursor(0, 30);
 oled.println("Spo2");

 oled.setTextSize(1);
 oled.setTextColor(1);
 oled.setCursor(0,45);
 oled.println(pox.getSpO2());
 oled.display();
```

Figure 5.11: Code for the OLED to Display Reading

Display Code

e. Code to Calculate Heart Rate, and Oxygen Saturation for MAX 30100 Sensor

The “BPM” and “SpO2” global variables need to be declared in the void loop. Then declare “PulseOximeter pox” from the library of MAX 30100 sensor that set the data sent in milliseconds. By using “getHeartRate()” and “getSpo2” that come from the library to get the data to calculate data and get data from the sensor before sent it to the NodeMCU to display on OLED and Blynk application.

```
pox.update();
Blynk.run();
timer.run(); // Initiates BlynkTimer

BPM = pox.getHeartRate();
Sp02 = pox.getSp02();
```

Figure 5.12: Code to Calculate Heart Rate, and Oxygen Saturation for MAX 30100 Sensor

Library Code

Calculation Code

f. Code for Heart Rate, and Oxygen Saturation

The code above, taken from Figure 5.13, demonstrates how the heart rate and oxygen saturation sensors will function in the future. As soon as the MAX 30100 sensor receives information from the user's body about heart rate and oxygen saturation, it will compute utilizing pulse oximeter libraries and then show the actual temperature on the Blynk application and the OLED display, respectively. The code shows the calculation of heart rate and oxygen saturation samples to get average data to help the user calculate in one minute. Then it needs to gain 60 samples of heart rate and oxygen saturation data.


```
void loop()
{
    pox.update();
    Blynk.run();
    timer.run(); // Initiates BlynkTimer

    BPM = pox.getHeartRate();
    Sp02 = pox.getSp02();
    if (millis() - tsLastReport > REPORTING_PERIOD_MS){
        sampleCount++;
        if (BPM > 0 && Sp02 > 0) {
          BPMTotal = BPMTotal + BPM;
          Sp02Total = Sp02Total + Sp02;
        } else {
          BPMTotal = 0;
          Sp02Total = 0;
          sampleCount = 0;
          BPMAvg = 0;
          Sp02Avg = 0;
        }
        tsLastReport = millis();
    }
```

Figure 5.13: Code for Heart Rate, and Oxygen Saturation

Calculation Average Code

g. Code to Connect NodeMCU to Blynk Application

In a void setup, the Blynk must begin before they run in the void loop. The beginning Blynk must include the auth token, Wi-Fi SSID, Wi-Fi password to connect the Blynk server responsible for all the communications between the smartphone and hardware. After that, the Blynk must be run to function as well. The code "Blynk.run()" and "Blynk.begin()" have been stored in the Blynk library.

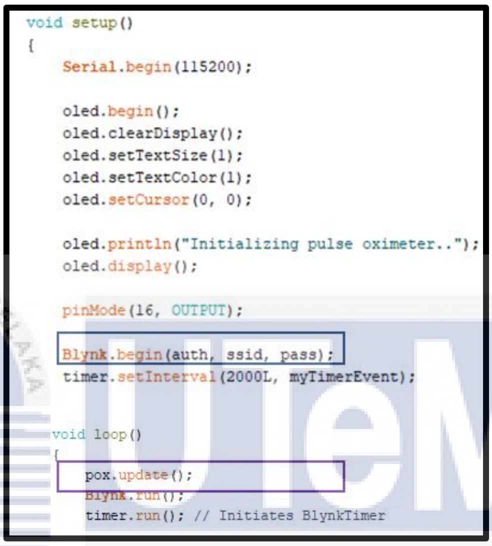


Figure 5.14: Code to Connect NodeMCU to Blynk Application

Begin Code

Run Code

h. NodeMCU Setup Coding

Figure 5.15 shows the setup code for the Node-MCU microcontroller. When the Node-MCU is turned on or connected to a PC, this setup will run automatically. The code will upload to the Node-MCU microcontroller and check for acceptable credentials for the Blynk token, Wi-Fi name, and Wi-Fi password; if all the credentials are correct, the serial monitor, and OLED display will display. "Initializing pulse oximeter...", and the project will function successfully.

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

    oled.begin();
    oled.clearDisplay();
    oled.setTextSize(1);
    oled.setTextColor(1);
    oled.setCursor(0, 0);

    oled.println("Initializing pulse oximeter..");
    oled.display();

    pinMode(16, OUTPUT);

    Blynk.begin(auth, ssid, pass);
    timer.setInterval(2000L, myTimerEvent);

    Serial.print("Initializing Pulse Oximeter..");
```

Figure 5.15: Setup Coding

i. Code to Threshold and Notification

Figure 5.16 illustrates the notification coding when the user's average SpO2 is less than 95%, and the heart rate is more than 140 per cent. The code will collect data from the user's SpO2 and heart rate to generate an average for the notification feature. Finally, it will send an automated notice to the user's smartphone through the Blynk application.

```
if (SpO2Avg > 0 && SpO2Avg < 95 && flagSpo2 == 0) {
  Blynk.notify("Low oxygen level in blood!");
  flagSpo2 = 1;
}
else if (SpO2Avg >= 95) {
  flagSpo2 = 0;
}

if (BPMAvg <= 0 && BPMAvg > 140 && flagBPM == 0) {
  Blynk.notify("Heart Rate Alert!");
  flagBPM = 1;
}
else if (BPMAvg <= 140) {
  flagBPM = 0;
}
```

Figure 5.16: Code to Threshold and Notification

Threshold Code

Notification Code

5.3.2 Software Configuration Setup

In this section, representation of how the data will be preview and monitor using Blynk will be explained in detail. This section also shows how the device are connect and communicate through Blynk.

i. Blynk Application

Readings from the sensor will be display on Blynk. Blynk is a platform that allows the user to control and monitor IoT projects with a quick setup. The platform used to monitor the output is on smartphone and Android or IOS devices. The configuration of Blynk application will be shown below:

1. Download the Blynk Application on a smartphone from Google Play Store or App Store.

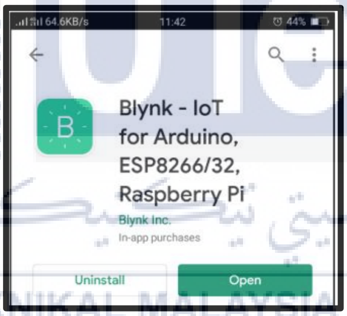


Figure 5.17: Blynk Application

2. Open the application once downloaded and create an account. Create a new project once the account is created.

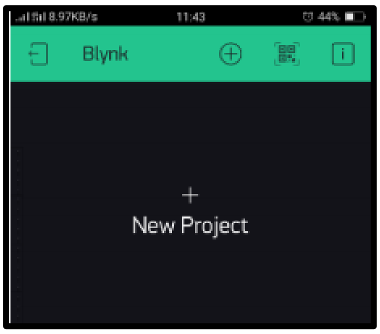


Figure 5.18: Create Project Interface

3. Once the project is created, there will be an authentication token sent to the registered email. The token is used for connection with the device.

4. The empty dashboard will be displayed, and widgets boxes are provided by clicking the icon on the top right of the application.

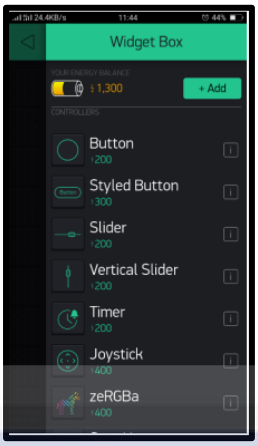


Figure 5.19: Widget Box

5. The complete design of the project on Blynk application.

Figure 5.20 Pulse Oximeter Interface

5.4 Complete Prototype

5.4.1 Prototype

Figure 5.21 shows the system architecture where the user may store the data from the Blynk application in an a.csv file. The sensor's oxygen saturation (SpO2) and heart rate (BPM) will be stored in the Blynk database. The system will automatically erase the output (data.csv) after the user is done with it, but it may be copied using the save as method. The prototype, finger or any other arterial pulse that can detect oxygen saturation (SpO2) and heart rate must be present to establish this scenario (BPM). After setting up the hardware, the Node-MCU connects to the Blynk application and sends all data to the Blynk server.

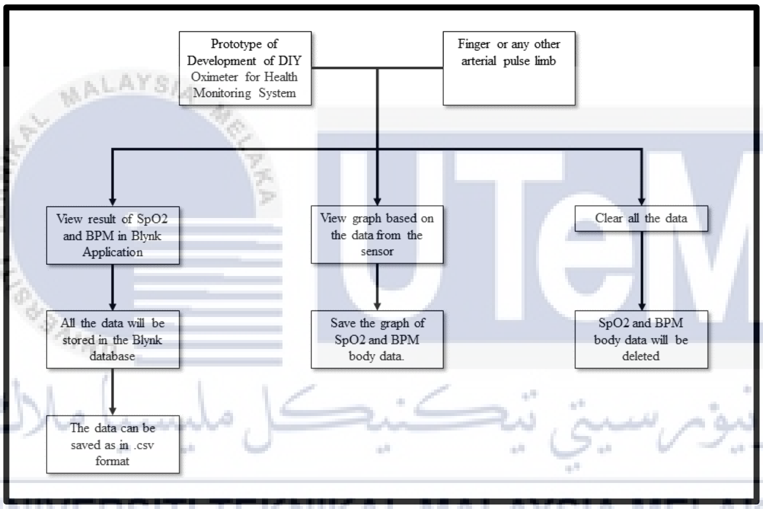


Figure 5.21: Architecture of Development of DIY Oximeter for Health Monitoring

The hardware that has been utilized in this project has previously been mentioned in Chapter 4. A few jumpers are connected, a MAX 30100 sensor, an OLED display, and a breadboard are utilized, and will connect all this hardware to the microcontroller, a Node-MCU board. The Node-MCU board will accept commands from the Arduino IDE software program to control any sensors and displays connected to the breadboard and Node-MCU board. Figure 5.22 below illustrates the specific pins that will utilize to link the sensor and LCD to the Node-MCU board through the breadboard.

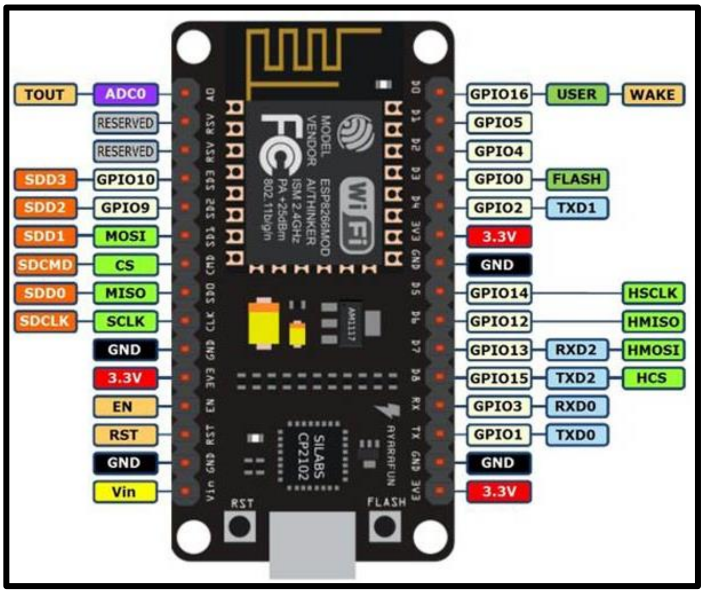


Figure 5.22: Node-MCU Board Pins Details

Table 5.1 above details which pins will be utilized by hardware components such as the MAX 30100 sensor and an OLED display. Apart from the pins, this table 5.1 will describe the cables utilized by the sensor and LCD, as Chapter 4 did.

Table 5.1: Detail Pins Number for Node-MCU

Hardware	Wire	Pins
MAX 30100 Sensor	GND	GND
	INT	D0
	SDA	D2
	SCL	D1
	VIN	3V3
OLED Display	VCC	3V3
	GND	GND
	SCL	D1
	SDA	D2

Figure 5.23 shows the idea for a prototype that uses a MAX 30100 sensor and an OLED display connected to the Node-MCU ports to show real-time monitoring of oxygen saturation and heart rate. This prototype comes with all the required hardware. Hardware components, the Node-MCU board, are connected to the PC through the USB port, as indicated in the diagram.

Figure 5.23 shows the project prototype that is currently being developed. A few jumper wires for connection, a MAX 30100 sensor, an OLED display, a breadboard, and a NodeMCU are included in this prototype. Table 5.1 governs all connections. It will provide a better understanding of how the Development of DIY Oximeter for Health Monitoring with Smartphone Notification works in the actual world.

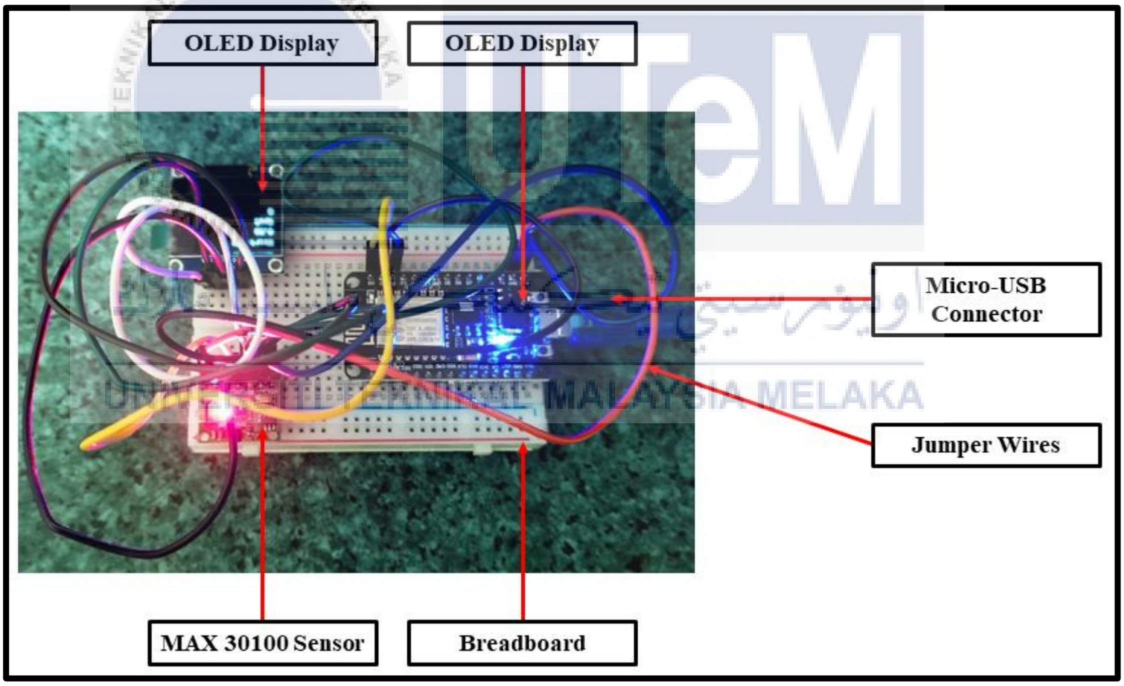


Figure 5.23: View of The Prototype

5.4.2 Software

Figure 5.24 shows the interface page for the pulse oximeter system on Blynk application.

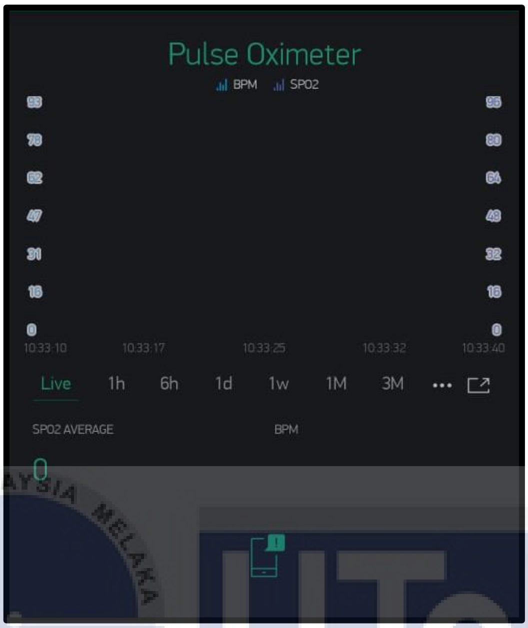


Figure 5.24 Interface Pulse Oximeter Application

5.4.3 Integration Between Prototype and Software

In this section, it will be explained in detail the development of this project system by using the figure 5.25 below.

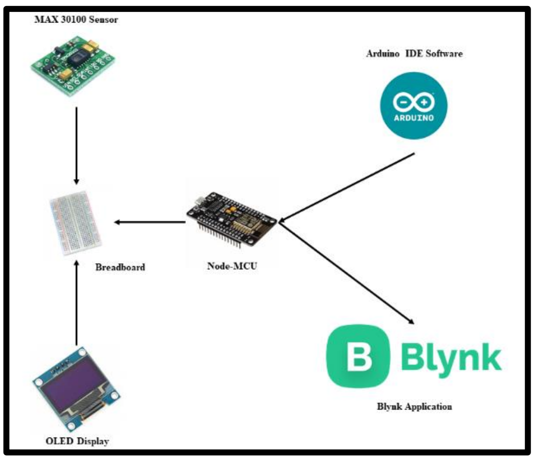


Figure 5.25: Integration Between Prototype and Software

The system deployment shown in Figure 5.25 is for developing a DIY Oximeter for Health Monitoring system. The Arduino IDE software serves as a platform for writing code that controls the LCD and sensor. The Arduino IDE software reads and displays the sensor's data through the serial monitor. The Node-MCU accepts the Arduino IDE software code through a serial connection such as COM 3 or COM 5. Then, by implementing the Blynk code in the Arduino IDE program, all MAX 30100 will be transmitted to the Blynk server and saved in the Blynk database through the Node-MCU.

Figure 5.26 below shows integration between prototype and pulse oximeter system on Blynk application.

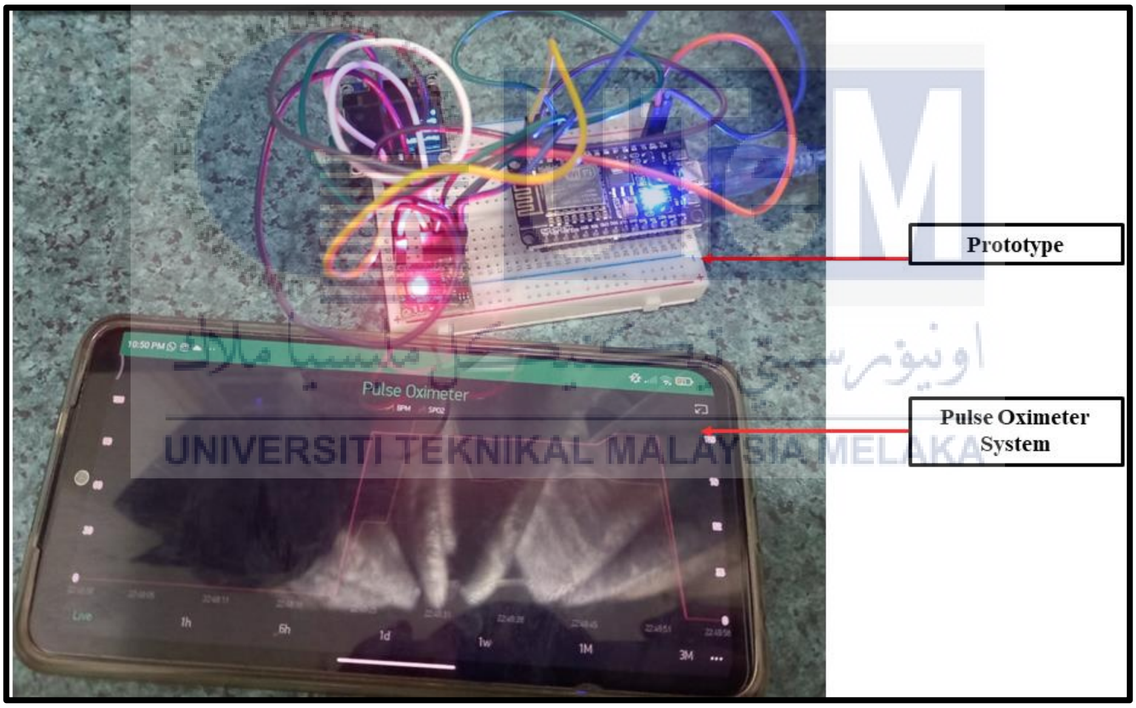


Figure 5.26: Real Environment Integration Between Prototype and Software

5.5 Implementation Status

In this part, the duration for each component to development, and complete will be discuss and explain the details of development of project in this part.

Table 5.2: Implementation Status

No	Component	Description	Duration for completed
1.	Assembling Hardware	➢ Obtaining all the required components or hardware for the project. ➢ Process to connect jumper wires, MAX 30100 sensor, breadboard, and Node-MCU. ➢ Board by assigning the pins on the USB port of laptop.	14 days
2.	Building Prototype	➢ This process is to setup the prototype of the project.	10 days
3.	Coding, Compile, and Upload Arduino Uno Software Coding	➢ This process is to ensure Arduino microcontroller can control the hardware attached to it normally.	8 days
4.	Source Code Implementation	➢ Process of creating source code to make the prototype functional.	14 days
5.	Developing Blynk Application	➢ All the data will be displayed in Blynk application and can be saved using excel (.csv).	10 days

5.6 Conclusion

In conclusion, the implementation chapter is the most critical for implementing a system since it details the hardware configuration, software installation, and development processes. It establishes a clear vision and strategy for developing the project to accomplish the project's goal. Additionally, it is the most critical component of our project since we must build the project prototype and implement all necessary code to ensure that the system operates as intended. Finally, the next chapter will cover the project's testing phase. The testing phase will cover various topics, including the examination of all sensors connected to the Node-MCU.

CHAPTER 6: TESTING AND ANALYSIS

6.1 Introduction

In the last chapter, we covered the project's implementation briefly. After completing the implementation phase of this project, we will now discuss and evaluate the testing phase. Testing is one of the stages to ensuring that the project is finished and meets the goals, including easy development of a DIY oximeter for health monitoring system, alarm notification, data recording, and user-friendliness. This section will describe the project's results and analyses by giving a graphical representation of the data gathered during testing. This chapter will demonstrate the connection between the smartphone Blynk application, the MAX 30100 sensor, and the Node-MCU microcontroller. This chapter summarizes the findings of the analysis.

6.2 Test Plan

This section will discuss the fundamentals of each project's system testing. This section will also describe the testing scope and the activities that will occur throughout the testing phase. It is critical to guarantee that all objectives are met, and that the system operates smoothly. Any flaws or mistakes may be identified and corrected before the application is sent to the user.

6.2.1 Test Organization

This section will include two individuals responsible for testing the system. The first is the system developer, who creates the system, and the second is the end-user, who will use the project's functioning. Each tester has a unique function to perform throughout the testing process.

a) **System Developer**

The system developer is responsible for testing the system, looking for flaws or errors. Additionally, the system developer is accountable for the system's stability and seamless operation.

b) **End User**

The end-user is responsible for finding out any system weakness, giving feedback regarding interface, functionality, and way to enhance the design. The end-user will be an adult or child to test the function of the system.

6.2.2 Test Environment

The testing environment will be dictated by the location and preparation of the project. The area of the project must be provided to calculate device limitations. Node-MCU must be used to link the device and Blynk application. The collected data must be monitored and handled to detect any device malfunctions. Before testing, the Node-MCU Microcontroller and sensor will be connected via USB connection to a PC or laptop. The Node-MCU software will be run on a Windows 10 laptop linked to the internet. Is to ensure that the real-time notification system functions appropriately.

6.2.3 Test Schedule

This part will discuss how the system developer tests their system while the testing process is in progress. If any problems were discovered during testing, the system developer would return to the implementation phase to identify and fix any potential flaws. Once this procedure has been completed, it will be repeated indefinitely until the desired outcome has been achieved.

6.3 Test Strategy

In this section, we will explain how to decides the approach utilized to test the system. Black box testing is a method for evaluating the system's functioning. Black box testing is a method to assess the functioning of a project without delving into the project's underlying code structure or implementation details. The black box is primarily concerned with the project's input and output. The diagram of black-box testing is shown in Figure 6.1.

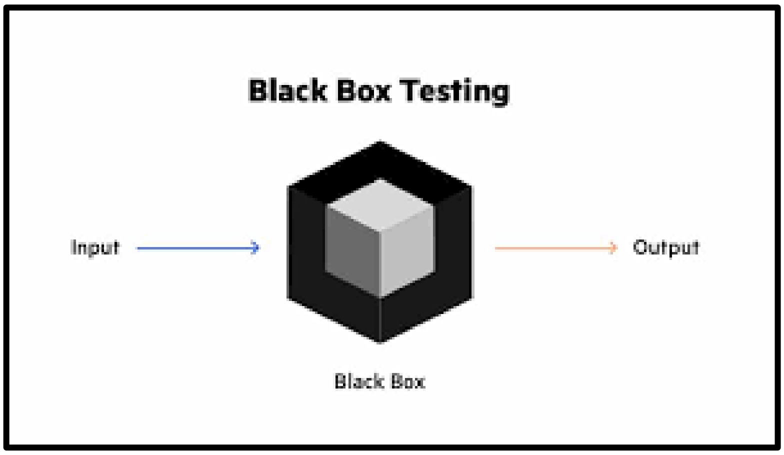


Figure 6.1: Black Box Testing (Imperva, Unknown**)**

Black box testing includes a variety of tests geared at functional, non-functional, and regression testing. This project will concentrate on functional testing to ensure that the project functions appropriately. The developer will use this approach to identify the applicable requirements for the project. The examination will begin with several sets of heart rate and oxygen saturation measurements. Will track heart rate and oxygen saturation readings will track heart rate and oxygen saturation readings through the Blynk application on a smartphone.

6.3.1 Classes of test

a) **Functionality Test**

The functionality test is performed to ensure that NodeMCU and the MAX 30100 sensor are in good functioning conditions. When the NodeMCU receives data from the MAX 30100 sensor and sends it to the Blynk program, the testing process may begin.

b) **Position Test**

The position test is performed to ensure the area to reading the heart rate, and oxygen saturation data to taken reading for the record and visualize data in Blynk application.

6.4 Test Design

Test design is concerned with the discovery of test cases, the development of test cases, and the prediction of anticipated results for each scenario that has been created and recorded. The integration test and the functionality test are discussed in detail in the test description.

6.4.1 Test Description

I. Connectivity Test between NodeMCU and Computer

Table 6.1: Connectivity of NodeMCU and Computer

Test Case ID	**TC_01**
Test Functionality	Connectivity test of NodeMCU and computer.
Precondition	- NodeMCU and computer are connected using USB adapter. - Arduino IDE must be installed in the computer.
Execution Steps	1. Launch Arduino IDE once computer have been turned on. 2. Connect NodeMCU to the computer by using USB cable. 3. Upload the C++ script into the NodeMCU board that is connected to the computer.
Expected Results	- Arduino IDE can detect the connection of NodeMCU board. - The C++ script is successfully uploaded into the NodeMCU board.
Error Message	None
Result	Pass

II. Connectivity Test Between Sensor and NodeMCU

Table 6.2: Connectivity Test Between Sensor and NodeMCU

Test Case ID	**TC_02**
Test Functionality	Connectivity test of NodeMCU and MAX30100 sensor.
Precondition	- NodeMCU and computer are connected using USB adapter. - NodeMCU and sensor are connected using jumper wire. - Arduino IDE must be installed in the computer.
Execution Steps	1. Launch Arduino IDE once computer have been turned on. 2. Connect NodeMCU to the computer by using USB cable. 3. Connect NodeMCU and MAX 30100 sensor by using jumper wire. 4. Upload the C++ script into the NodeMCU board that is connected to the computer. 5. Open serial monitor to observe any input from the sensor
Expected Results	- The MAX 30100 sensor is giving output to the NodeMCU board. - NodeMCU should be able to calculate and show the measurement of heart rate, and oxygen saturation in serial monitor. - The C++ script is successfully uploaded into the NodeMCU board.
Error Message	None
Result	Pass
Remark	The test must be completed in one or two minutes to get a stable heart rate and oxygen saturation because

	some users may have cold hands, their blood may flow slowly, or their blood pressure may be low or high during the reading, it is preferable if they take the reading for more than one minute.

III. Connectivity Test Between OLED and NodeMCU

Table 6.3: Connectivity Test Between OLED and NodeMCU

Test Case ID	**TC_03**
Test Functionality	Connectivity test of NodeMCU and OLED display.
Precondition	- NodeMCU and computer are connected using USB adapter. - NodeMCU and OLED display are connected using jumper wire. - Arduino IDE must be installed in the computer.
Execution Steps	1. Launch Arduino IDE once computer have been turned on. 2. Connect NodeMCU to the computer by using USB cable. 3. Connect NodeMCU and OLED display by using jumper wire. 4. Upload the C++ script into the NodeMCU board that is connected to the computer. 5. Open serial monitor to observe if any error for OLED display. 6. The OLED should show the reading of heart rate and oxygen saturation from MAX 30100 sensor.
Expected Results	- The NodeMCU is giving output of the MAX 30100 sensor to the OLED display. - The OLED display should the reading of heart rate and oxygen saturation from MAX 30100 sensor.

Error Message	None
Result	Pass

IV. Connectivity Test Between Blynk Application and NodeMCU

Table 6.4: Connectivity Test Between Blynk Application and NodeMCU

Test Case ID	**TC_04**
Test Functionality	Connectivity test of NodeMCU and Blynk application.
Precondition	- NodeMCU and computer are connected using USB adapter. - NodeMCU and OLED display are connected using jumper wire. - NodeMCU and sensor are connected using jumper wire. - Blynk application have been installed on smartphone.
Execution Steps	1. Upload the codes that have an integration with Blynk application on NodeMCU. 2. Open the serial monitor on Arduino IDE to observe the connection.
Expected Results	- NodeMCU and Blynk application are successfully connected based on the display on serial monitor.
Error Message	None
Result	Pass

V. Reading Test Based on the Location of Sensor

Table 6.5: Reading Test Based on Location of Sensor

Test Case ID	**TC_05**
Test Functionality	Reading test based on the on the location of the sensor on parts of hand.
Precondition	- Blynk application and NodeMCU are connected. - All the codes have been uploaded. - Blynk application have been installed on smartphone
Execution Steps	1. Put the sensor on different location which is on wrist and fingers. 2. Open the serial monitor on Arduino IDE and Blynk application on smartphone to observe the readings.
Expected Results	- To detect best location of the sensor to the detect heart rate, and oxygen saturation is obtained.
Error Message	None
Result	Pass
Remark	The test must be completed in one or two minutes to get a stable heart rate and oxygen saturation because some users may have cold hands, their blood may flow slowly, or their blood pressure may be low or high during the reading, it is preferable if they take the reading for more than one minute.

VI. Reading Test for The Accuracy

Table 6.6: Reading Test for The Accuracy

Test Case ID	**TC_06**
Test Functionality	Accuracy test of the MAX 30100 sensor
Precondition	- Blynk application and NodeMCU are connected. - All the codes have been uploaded. - Blynk application have been installed on smartphone
Execution Steps	1. Put the sensor on different location which is on wrist and fingers. 2. Open the serial monitor on Arduino IDE and Blynk application on smartphone to observe the readings. 3. Put smartwatch on wrist, oximeter on fingertips, or proposed solution on fingertips. 4. The testing perform separately. 5. Wait for 1 or 2 minutes to get the stable reading. 6. Compare the reading to get the results.
Expected Results	- An approximate equal reading from the techniques is obtains.
Error Message	None
Result	Pass
Remark	The test must be completed in one or two minutes to get a stable heart rate and oxygen saturation because some users may have cold hands, their blood may flow slowly, or their blood pressure may be low or high during the reading, it is preferable if they take the reading for more than one minute.

VII. Blynk Application Notification Test

Table 6.7: Blynk Application Notification Test

Test Case ID	**TC_07**
Test Functionality	Connectivity test of Blynk Application Notification Test.
Precondition	- NodeMCU and computer are connected using USB adapter. - NodeMCU must declare Auth Blynk. - Install Blynk Application on the smartphone. - Arduino IDE must be installed in the computer.
Execution Steps	1. Launch Arduino IDE once computer have been turned on. 2. Connect NodeMCU to the computer by using USB cable. 3. Upload the C++ script into the NodeMCU board that is connected to the computer. 4. Install Blynk Application on smartphone to test the notification. 5. Open serial monitor to observe any input from Blynk is connected.
Expected Results	- The notification will pop-up in the smartphone. - NodeMCU should be able to connect to the Blynk server in serial monitor. - The C++ script is successfully uploaded into the NodeMCU board.
Error Message	None
Result	Pass

VIII. Blynk Data Record Test

Table 6.8: Blynk Data Record Test

Test Case ID	**TC_08**
Test Functionality	Connectivity test of Blynk Application Data Record Test.
Precondition	- NodeMCU and computer are connected using USB adapter. - NodeMCU must declare Auth Blynk. - Install Blynk Application on the smartphone. - Add widgets SuperChat in Blynk Application - Arduino IDE must be installed in the computer.
Execution Steps	1. Launch Arduino IDE once computer have been turned on. 2. Connect NodeMCU to the computer by using USB cable. 3. Upload the C++ script into the NodeMCU board that is connected to the computer. 4. Install Blynk Application and add widgets SuperChart on smartphone to test the data record. 5. Open serial monitor to observe any input from Blynk is connected.
Expected Results	- The data can be record and sent to the email users.. - NodeMCU should be able to connect to the Blynk server in serial monitor. - The C++ script is successfully uploaded into the NodeMCU board.
Error Message	None
Result	Pass

6.4.2 Test Data

I. NodeMCU and Computer Connectivity Test

The connection between the NodeMCU and the PC is shown in Figure 6.2 using a USB cable. This connection is critical for the project to go forward to the next stage, uploading the code.

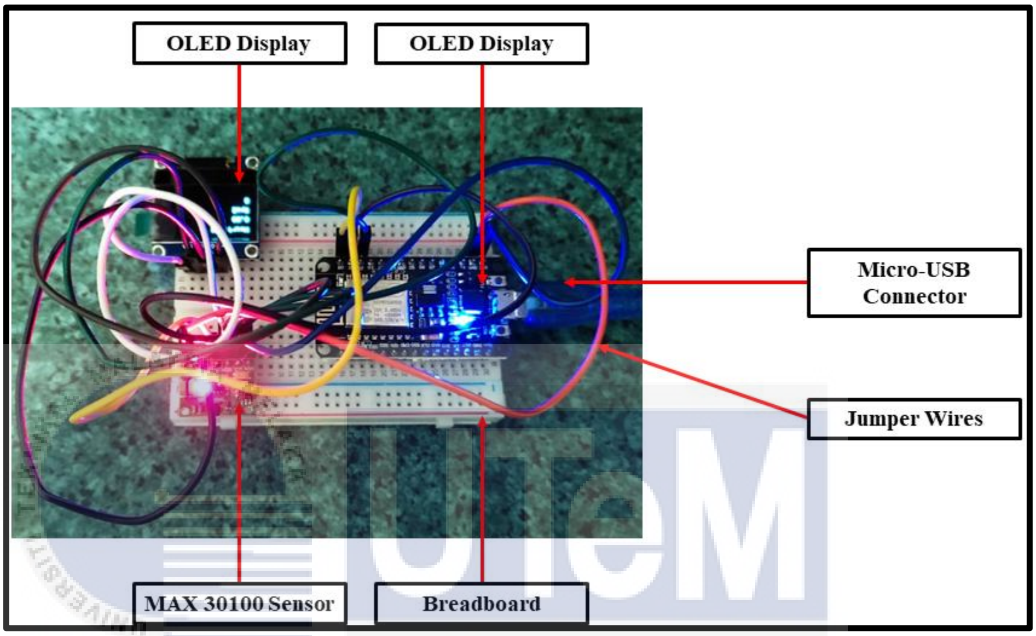


Figure 6.2: Connection Between NodeMCU and Computer

As shown in Figure 6.3, the Arduino IDE has recognised NodeMCU as a computer-connected board. This test verifies that NodeMCU has been connected successfully.

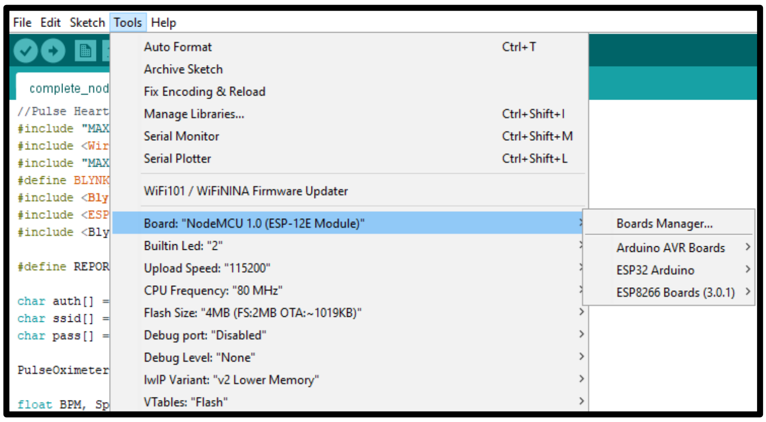


Figure 6.3: Board NodeMCU Function

After establishing a successful connection, upload the source code to NodeMCU. The console window indicates that the source code has been successfully uploaded to the NodeMCU board, as shown in Figure 6.4.

```
Compressed 287136 bytes to 209753...
Writing at 0x00000000... (7 %)
Writing at 0x00004000... (15 %)
Writing at 0x00008000... (23 %)
Writing at 0x0000c000... (30 %)
Writing at 0x00010000... (38 %)
Writing at 0x00014000... (46 %)
Writing at 0x00018000... (53 %)
Writing at 0x0001c000... (61 %)
Writing at 0x00020000... (69 %)
Writing at 0x00024000... (76 %)
Writing at 0x00028000... (84 %)
Writing at 0x0002c000... (92 %)
Writing at 0x00030000... (100 %)
Wrote 287136 bytes (209753 compressed) at 0x00000000 in 18.7 seconds (effective 122.6 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

Figure 6.4: Coding has been uploaded successfully

II. Sensor and NodeMCU Connectivity Test

The MAX30100 sensor connected to NodeMCU through the "VIN" pin to the "3V3" pin, the "SCL" pin to the "D1" pin, the "SDA" pin to the "D2" pin, the "INT" pin to the "D0" pin, and the "GND" pin to the "GND" pin. Refer to Chapter IV design and Chapter V implementation for a more comprehensive circuit schematic. The configuration for the linked sensor and NodeMCU is shown in Figure 6.5.

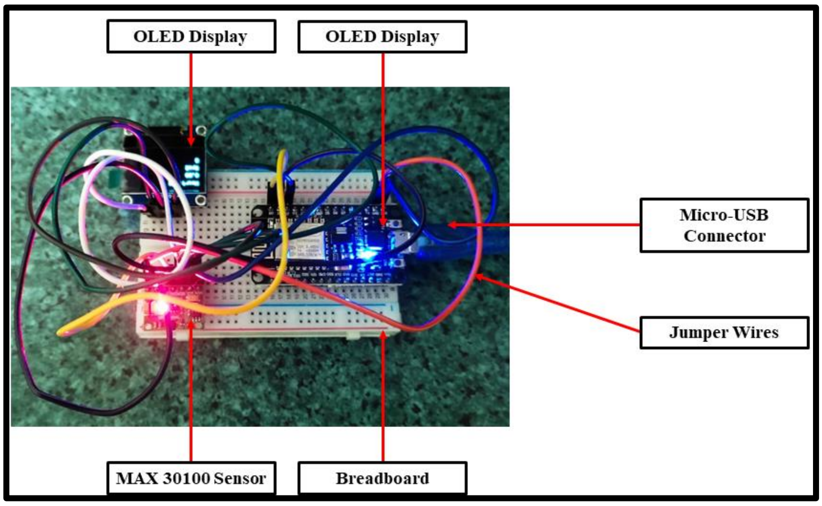


Figure 6.5: Connectivity NodeMCU and MAX30100 Sensor

Once the setup is complete, the NodeMCU board is uploaded with the code for monitoring the heart rate and oxygen saturation. When the code is uploaded, may use the serial monitor, and Blynk application to collect readings such as heart rate and oxygen saturation. After the computation is completed, it will take a few seconds before correct data is obtained. The monitor, and Blynk application will display the output in the manner shown in Figures 6.6 and 6.7 below.

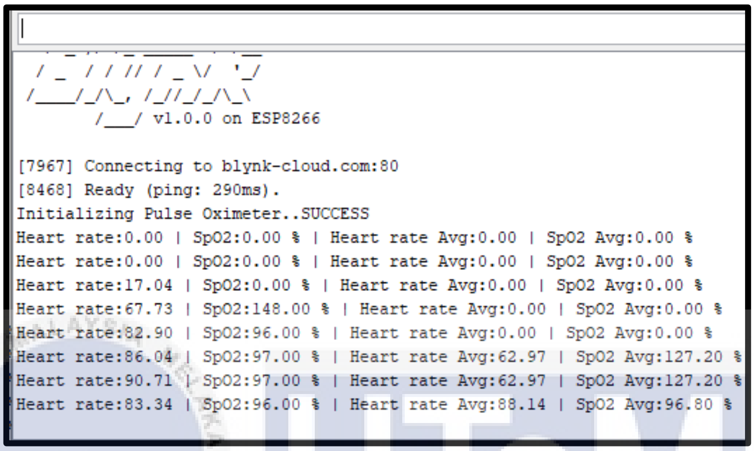


Figure 6.6: Output Reading Sensor from The Sensor on Serial Monitor

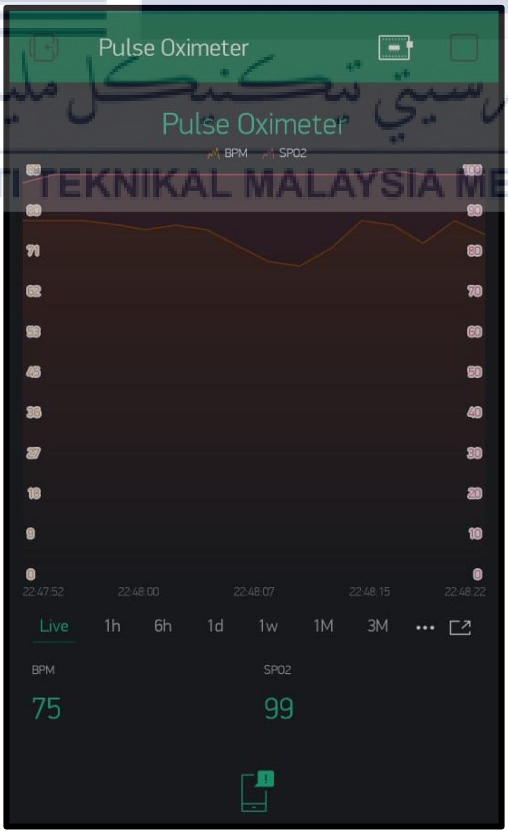


Figure 6.7: Output Reading Sensor on Blynk Application

III. OLED Display and NodeMCU Connectivity Test

The OLED Display sensor is connected to NodeMCU through the "GND" pin to the "GND" pin, the "VCC" pin to the "3V3" pin, the "SDA" pin to the "D2" pin, and the "SCL" pin to the "D1" pin. Refer to Chapter IV design and Chapter V implementation for a more comprehensive circuit schematic. The configuration for the linked sensor and NodeMCU is shown in Figure 6.7 below.

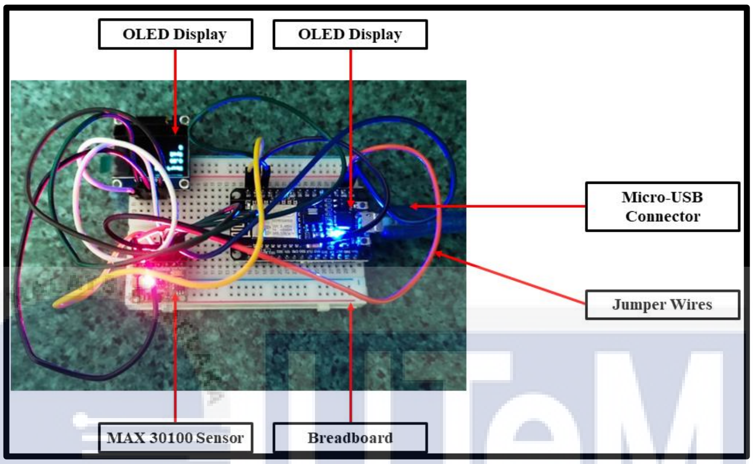


Figure 6.8: Connectivity NodeMCU and OLED Display Sensor

When the setup is complete, the code for measuring the heart rate is uploaded to the NodeMCU board. When the code is uploaded, the reading can be obtained by monitoring the heart rate and oxygen saturation on the OLED display. It will take a few seconds before valid data is received after a particular calculation is done. The monitor will show the output as shown in Figure 6.8 below.

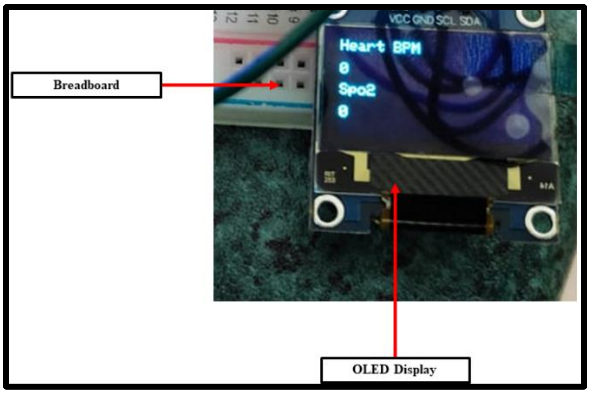


Figure 6.9: Output Reading Sensor from The Sensor on OLED Display

IV. Blynk Application Connectivity Test

NodeMCU is equipped with an ESP8266 Wi-Fi module in this project. It is essential to include an ESP8266 Wi-Fi module on the NodeMCU board to communicate with the smartphone over the internet and connect via the Blynk application. The output of the serial monitor is shown in Figure 6.9 above after the connection is established successfully. It indicates that Blynk is configured for the NodeMCU board in the manner specified during the Blynk application's interface development.

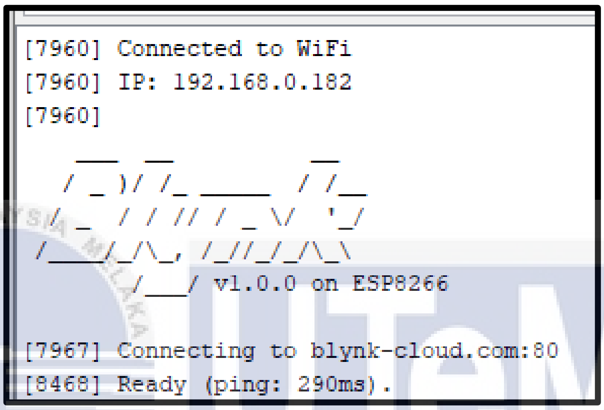


Figure 6.10: Connection of Blynk on Serial Monitor

V. Blynk Application Notification Test

When a designated threshold is surpassed, the Blynk application will alert the user, indicating that the oxygen saturation or heart rate is high. If the heart a user's heart rate is the value of 140 BPM, it will send an alert showing that the user is heart rate alert. If the user's oxygen saturation goes below 95 per cent, it will push a signal indicating that the user is a low oxygen level in blood. The coding "Blynk.notify" is coding to send notification or pop-up if the designated threshold value is surpassed. Figure 6.10 shows the code for threshold coding.

```
if (sampleCount >= 5) {
  BPMAvg = BPMTotal / sampleCount;
  SpO2Avg = SpO2Total / sampleCount;

  BPMTotal = 0;
  SpO2Total = 0;

  if (SpO2 > 0 && SpO2 < 95 && flagSpo2 == 0) {
    Blynk.notify("Low oxygen level in blood!");
    flagSpo2 = 1;
  }
  else if (SpO2 >= 95) {
    flagSpo2 = 0;
  }

  if (BPM <= 0 && BPM > 140 && flagBPM == 0) {
    Blynk.notify("Heart Rate Alert!");
    flagBPM = 1;
  }
  else if (BPM <= 140) {
    flagBPM = 0;
  }
```

Figure 6.11: Code Snippet for SpO2 and BPM Condition Notification

Threshold Code

Notification Code

Figure 6.11 illustrates the notice delivered to the smartphone. Additionally, it offers a sound warning to inform the user.

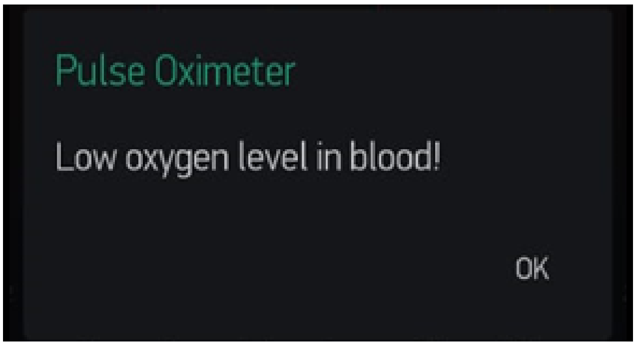


Figure 6.12: Notification Message on Smartphone

VI. Reading Test Based on The Position of The Sensor

The placement of the MAX30100 sensors is critical in this project since it determines the optimal location that will provide the most outstanding results. For position 1, the area is the fingertips; for position 2, the site is the right side of the wrist; and for position 3, the area is the left side of the wrist. You've selected this position to determine the most advantageous place for data gathering. The reason for choosing positions 2 and 3 is to ensure that the pulse or oxygen saturation measurements are correct in those places. In contrast to position 1, positions 2 and 3 are the best sites for monitoring the pulse manually. The prominent position 1 is well-suited for obtaining oximeter readings. The visual representation of the sensor location is shown in Figure 6.12.

Figure 6.13: Position of Wrist and Fingertips

Figure 6.13 show the testing set-up for the prototype. Will place a position wrist and fingertips will place on the sensor to get the data. Figure 6.14 show the block diagram of this project in the top view of the set-up.

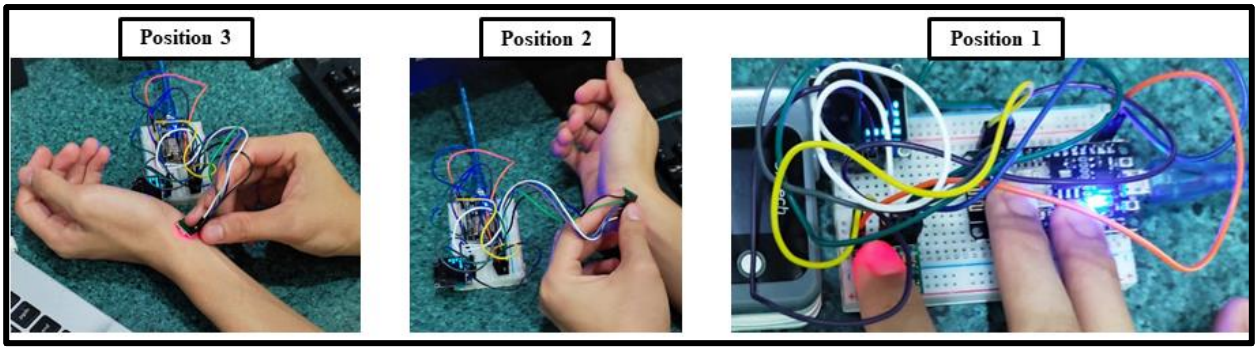


Figure 6.14: Position Setup for Testing

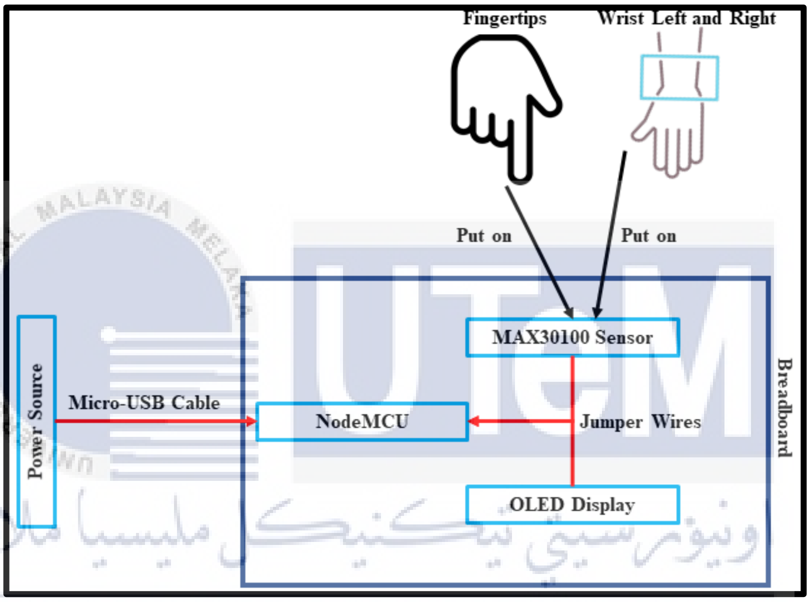


Figure 6.15: Block Diagram from Top View of The Setup

Table 6.9: Position Test of Heart Rate Measurement

Position	**Test Reading (BPM)**	**Time taken to obtain reading (m)**
Position 1	75	1 minutes
	73	1 minutes
	74	1 minutes
Average Position 1	**74**	
Position 2	0	1 minutes
	0	1 minutes
	0	1 minutes
Average Position 2	0	

Position 3	0	1 minutes
	0	1 minutes
	0	1 minutes
Average Position 3	0	

Table 6.10: Position Test for Oxygen Saturation (SpO2)

Position	**Test Reading (SpO2)**	**Time taken to obtain reading (m)**
Position 1	99%	1 minutes
	99%	1 minutes
	99%	1 minutes
Average Position 1	99%	
Position 2	0%	1 minutes
	0%	1 minutes
	0%	1 minutes
Average Position 2	0%	
Position 3	0%	1 minutes
	0%	1 minutes
	0%	1 minutes
Average Position 3	0%	

From tables 6.9 and 6.10, the test reading is where the heart rate and oxygen are obtaining by using this proposed solution device. Each position is compared with the time taken to get the reading of heart rate and oxygen saturation. Position 1 can detect the heart rate by using the MAX30100 sensor. Position 2 and 3 can detect the heart rate but the reading of heart rate and oxygen saturation is unstable compared to the position after 1 minute. The stable reading from time to obtain a reading for the sensor to detect heart rate and oxygen saturation determines that the position can provide the heart rate and oxygen saturation effectively. Finding the best position to put the sensor is significant in providing the best result.

VII. Reading Test for The Accuracy

This test is done to measure the accuracy of the data when measuring the heart rate and oxygen saturation. It can also measure the heart rate by calculating the pulse per minute using the commercial product and proposed solution, and the same goes with oxygen saturation. The method to do it is by putting a commercial pulse oximeter on the fingertips. Wait for 1 minute to get stable data. Figure 6.13 shows how the technique is done.

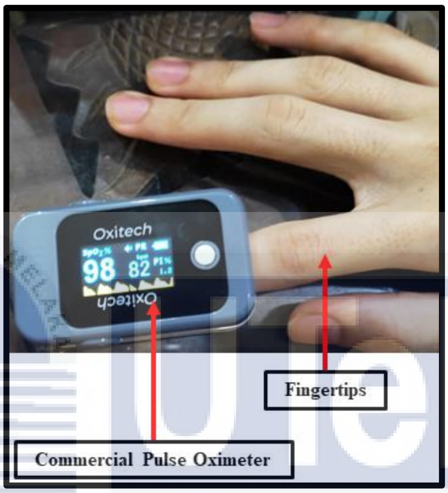


Figure 6.16: Commercial Pulse Oximeter

Table 6.11 shows that the measurement results of the SpO2 percentage have an average error (error) with the standard commercial equipment is about 3%. The measurement results of the SpO2 rate have an average error (error) with the standard commercial equipment is about 0.3%. The error percentage means that the proposed solution oximetry is comparable with the commercial one. It shows that both methods are approximately similar reading.

Table 6.11: Comparison between Two Methods of Calculating Heart Rate and Oxygen Saturation

Reading	Commercial Pulse Oximeter		Proposed Solution	
	SpO2	BPM	SpO2	BPM
1	99	81	100	81
2	100	78	100	76
3	99	95	99	88
Average	99.3	84.6	99.6	81.6
Error	99.6 – 99.3 = 0.3		84.6 – 81.6 = 3	

VIII. Blynk Data Record Test

Sensor data is essential in any project to find how the system performs or monitor a system. In this project, the data can plot sensor data charts from the cloud. This project doesn't need to be on the same Wi-Fi network. Users can access sensor data from anywhere in the world. Users not only see how to plot sensor data but also export this data in a CSV file. All these things will be going in Blynk App. This project needs to add widgets SuperChart on dashboard Blynk Application based on Figure 6.17, showing how to add the widgets. The benefits of using SuperChart the data can be viewed in the graph for real-time monitoring and save data in a CSV file based on data in the graph using the SuperChart.

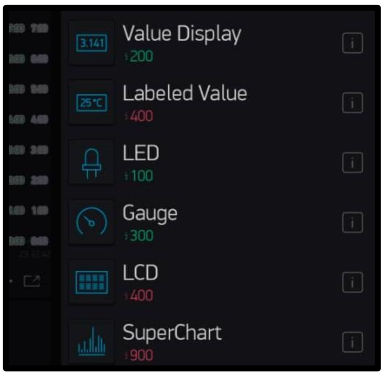


Figure 6.17: Widgets of SuperChart

Figure 6.18 shows the options users can choose if they want to import the data in CSV sent through email or erase the data. The users will get the data Pulse Oximeter v7 for BPM and Pulse Oximeter v8 for SpO2 on email based on Figure 6.19.

Figure 6.18: Option of SuperChart

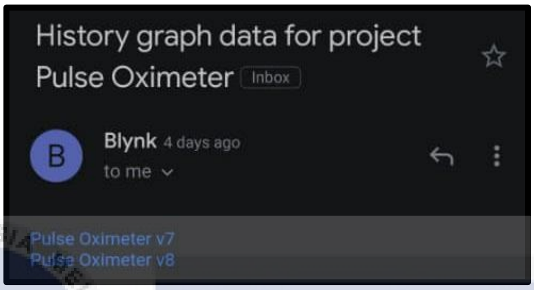


Figure 6.19: Email of Pulse Oximeter

The heart rate and oxygen saturation data will save based on the graph if the users do not erase the data. The duration of the previous Blynk application can keep four months after the developer test to import data based on Figure 6.20 for the early month's data saved on the CSV file and Figure 6.21 for the latest data held on the CSV file. The data will give the timestamp, not the date and time. They have two to convert the data using the online conversion method based on Figure 6.22 or a formula based on Figure 6.23. The conversion of this timestamp to make it users understand when the data have been recorded and readable for human.

1	30.9206	25/5/2021 16:57	0
2	30.93056	25/5/2021 16:58	0
3	30.93556	25/5/2021 16:59	0

Figure 6.20: Early Data

3047	60.10083	3/9/2021 19:44	0
3048	60.23117	3/9/2021 19:45	0
3049	60.14467	3/9/2021 19:46	0

Figure 6.21: Latest Data

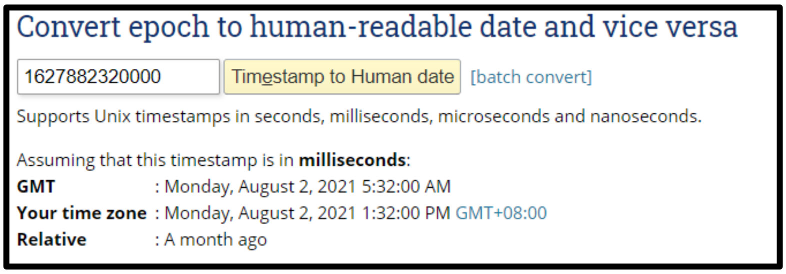
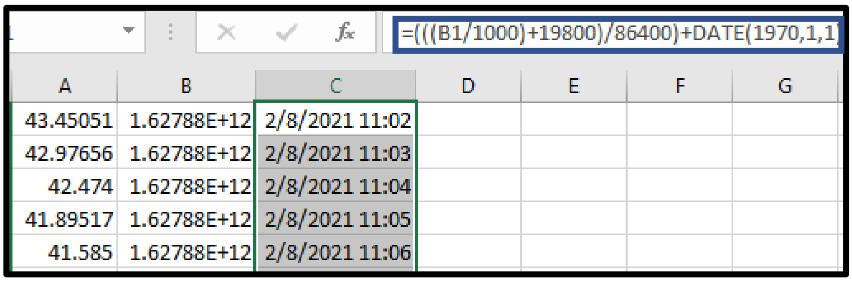


Figure 6.22: Online Conversion



Formula

Figure 6.23: Formula Conversion

6.5 Test Result and Analysis

6.5.1 Accuracy of Sensor and Comparative User Analysis Testing

This analysis aims to test MAX30100 sensors compared to get the accuracy for this project when detecting heart rate and oxygen saturation based on the different devices. The MAX30100 sensor combines two LEDs, optimized optics, and low-noise analog signal processing to detect pulse oximetry and heart-rate signals. This test is done to measure the accuracy of the MAX30100 sensor when measuring the heart rate and oxygen saturation. The heart rate and oxygen saturation can also measure the heart rate and oxygen saturation by calculating the beats per minute (BPM) and peripheral capillary oxygen saturation (SpO2) using the smartwatch and commercial pulse oximeter in the market. Brand 1 used in this test is Huawei Band 5, which builds in with the heart rate and oxygen saturation to get the results for these two readings. Brand 2 used in this test is Oxitech with a heart rate and oxygen saturation sensor to get the result for this two-reading using the position one technique. For Brand 3 used is proposed solution, the reading is taking like the Oxitech with using position one then need to wait 1 or 2 minutes to get the stable reading. For the full specification of Brands 1 and 2, refer to Appendix A, and for Brand 3, refer to Chapter II.

This analysis also will be accurate different the user to get more accuracy of the sensor that also involved Brand 1, Brand 2, Brand 3. The involved user in this test analysis is three users from different age, skin, gender, and others. User 1, User 2, and User 3 may be different readings of heart saturation and oxygen saturation because different ages, skin, gender, and others can be factors. For example, someone over the age of 70 may have an oxygen saturation level of about 95%, considered appropriate (Lapum, J. L., Verkuyl, M., Garcia, W., St-Amant, O., & Tan, A., 2019). The National Health Service (NSH) in the United Kingdom has issued a warning that an oximeter. Which is used to monitor oxygen levels in the blood of The National Health Service (NSH) in the United Kingdom has issued an oximeter warning. Which is used to monitor oxygen levels in the blood of COVID-19 patients at home, may provide less accurate readings to darker skin patients.

This test will take three reading a day to get the system consistency and ideally start tracking from the baseline and record at least three times a day to keep constant monitoring for all three Brands. Record the data from the first until the third heart rate value and oxygen saturation within five days. The time for the reading is at 10.00 am, evening at 5.00 pm, and night at 9.00 pm to collected data to observe variations in readings at various times.

a) Environment Setup

Figure 6.24 shows the setup of NodeMCU, the MAX30100 sensor, fingertips, OLED display and all brands of this project. Figure 6.25 shows Brand 1, and figure 6.26 shows the setup of Brand 2. For the figure 6.27, it will show the block diagram for design all the brand.

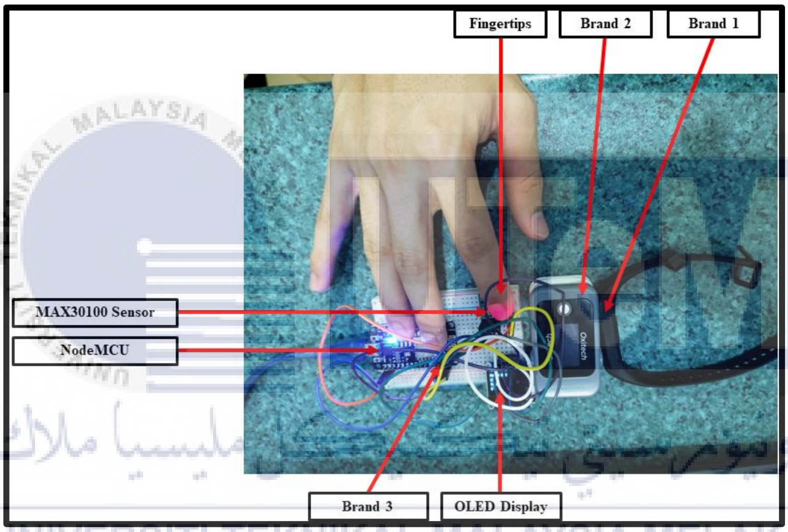


Figure 6.24: Setup Brand 3 for Accuracy and Comparative Test

Figure 6.25: Setup Brand 1 for Accuracy and Comparative Test

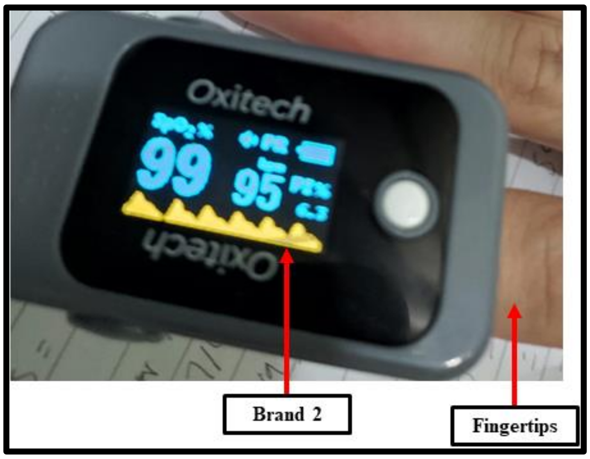


Figure 6.26: Setup Brand 2 for Accuracy and Compared Test

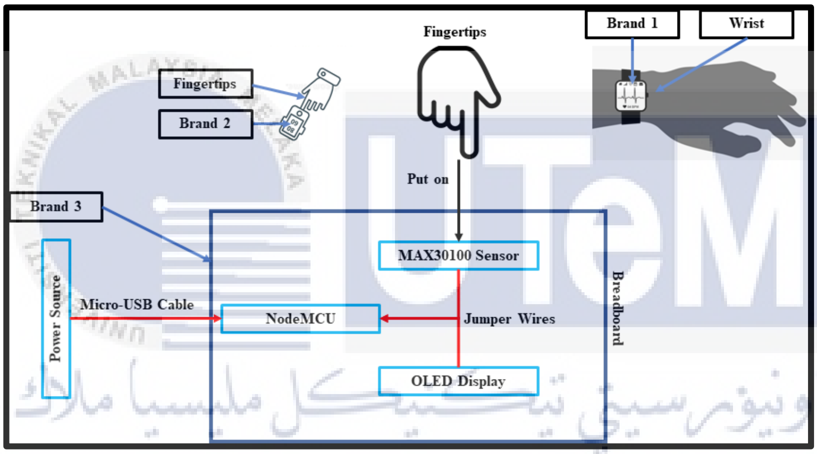


Figure 6.27: Block Diagram Setup All Brand

6.5.1.1 Scenario: Day 1 to Day 5 for Accuracy and Comparative Testing

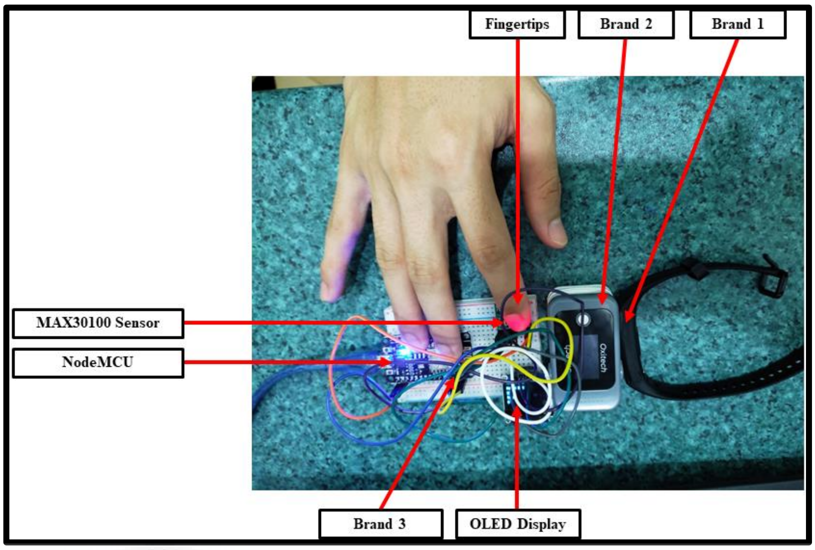


Figure 6.28: Setup for Accuracy and Comparative Testing

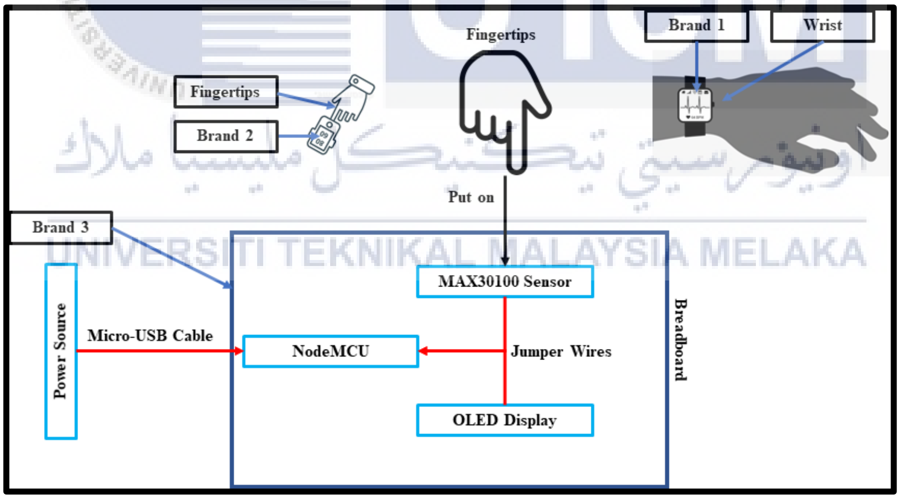


Figure 6.29: Block Diagram Setup for Accuracy and Comparative Testing

Test Case:

In this test case, the measure of heart rate and oxygen saturation based on the data from the three brand that can refer to the Appendix A and the three users from different details that can refer to the Appendix B, followed by these characteristics:

i. Record the data at least three times a day to monitor all three brands from the serial monitor or OLED display.
ii. Record the data from at least three different users to get the system consistency. Ideally, start tracking from the baseline and record at least three times a day to keep constant monitoring for all three brands.
iii. Place the sensor for brands 2 and 3 on the fingertips of the users.
iv. Place the sensor for brand one on the wrist of the users.
v. The testing performs separately.
vi. Take reading in 1 minutes from all users and all brands.

Table 6.12 shows day 1 of heart rate measurement data for three different users and three brands. For the total result of gathering data from day one to day five, refer to Appendix A. Data for the three-user specification can refer to in Appendix B. The first until third data is display on the serial monitor, OLED display, and smartphone for brand 3. The data for user one using these three brands have a similar average from 82.3, 84.6, and 81.6 because the reading data 3 times and average almost the same. Next, the data for user two using these three brands has different averages from 88, 90.6, 81.6 because the brand three evening data reading is slightly further than brand one and brand 2, and the average data for brand two is a little different compared to the brand one and brand 3. Besides that, the data for user three using these three brands has another average from 92.6, 87.6, and 86.3 because the average data for brand one is a little different compared to brand one and brand 3. Finally, the value is similar from these three brands to the reading for the three users because only two averages have above 90 per cent, and the average heart rate measurement data reading is above 80 per cent.

Table 6.12: Accuracy and Comparative Heart Rata Measurement Data

Brand	User	Morning (BPM)	Evening (BPM)	Night (BPM)	Average / 3 Times
1	1	79	79	89	**82.3**
	2	82	93	89	**88**
	3	84	99	91	**92.6**
2	1	81	78	95	**84.6**
	2	84	99	89	**90.6**
	3	80	91	92	**87.6**
3	1	81	76	88	**81.6**
	2	84	63	98	**81.6**
	3	79	91	89	**86.3**

Table 6.13 shows day 1 of oxygen saturation measurement data for three different users and three brands. For the total result of gathering data from day one to day five, data for the three-user specification can refer to in the Appendix A, and Appendix B. The first until third data is display on the serial monitor, OLED display, and smartphone for Brand 3. The data for user one using these three brands have a similar average from 98.6, 99.3, and 99.6 because the reading data 3 times and average almost the same for these three users. Next, the data for user two using these three brands has a similar average from 98, 99.6, 97 because the reading data 3 times and average almost the same for these three users. Besides that, the data for user three using these three brands has a similar average from 95.6, 97.6, and 97.6 because the reading data 3 times and average almost the same for these three users. Finally, the value is similar from these three brands to the reading for the three users, and the oxygen saturation data reading is above 90 per cent.

Table 6.13: Accuracy and Comparative Oxygen Saturation Data

Brand	User	Morning (SpO2)	Evening (SpO2)	Night (SpO2)	Average
1	1	99%	98%	99%	**98.6**
	2	97%	99%	98%	**98**
	3	97%	97%	93%	**95.6**
2	1	99%	100%	99%	**99.3**
	2	100%	100%	99%	**99.6**
	3	98%	99%	96%	**97.6**
3	1	100%	100%	99%	**99.6**
	2	96%	97%	98%	**97**
	3	99%	96%	98%	**97.6**

As a minor observation, after gathering data for the heart rate measurement (BPM) and oxygen saturation (SpO2) from day 1 to day five, refer to Appendix C and Appendix D, and user details can refer to Appendix B. The accuracy of MAX 30100 of the proposed solution or brand 3 for heart measurement is very close values and average from day 1 to 5 for user 1, user 2, and user 3 compared with brand 1 and 2 while knowing that these two is certified and tested. However, the data and average for heart measurement of user 2 in days 1 and 5 using brand three are far from brand 1 and 2 because the reading is inconsistent when user 2 put the fingertips on the MAX30100 sensor. For the comparative user test, when test to the user 3 for heart measurement is high, and oxygen saturation is low after five days reading with three times taken reading because the age of user 3 one of a factor can go to the Appendix B for complete details. After all, the heart rate is high, and oxygen saturation is low except for the genetic and the skin compared to users one and two. The heart rate and oxygen saturation are very close values and average after five days for all users.

In conclusion, the age factor affecting older adults typically has lower oxygen saturation levels than younger adults. For example, someone older than 70 may have an oxygen saturation level of about 95%, which is an acceptable level. The reading of user 3 using the three brands and the brand can be taken as the measurement reference. Thus, it can safely deduce that the proposed system or brand 3 using the NodeMCU platform and MAX30100 sensor is accurate enough for further use to measurement the heart rate and oxygen saturation data.

6.6 System Usability Scale (SUS)

The System Usability Scale (SUS) is a collection of questionnaires designed to gather user input (Assistant Secretary for Public Affairs, 2013). The developer will inform all attendees of the system. Then the participant will fill out questions regarding the developer's project. The demo may be done utilizing online platforms on the Internet. SUS provides statistics to developers about how end-users feel about their projects, whether they are satisfied or require improvement.

We will construct a Google Form for the query to provide this system usability scale to a random user. After a few days, 12 respondents provided feedback and documented a DIY Oximeter for Health Monitoring. The 12 respondents have chosen from the developer from friends, and family to complete the form that have been give the link of Google Form in the group or personal messages. Two kinds of testing are performed on the input: the prototype's user interface and its functionality. The testing is done from the end-user's perspective, who will look the end-user interface, Blynk to view the data. Due to the Covid-19 pandemics, testing is conducted through an online meeting platform. The developer will be briefed on this prototype before doing end-user testing. After testing is complete, the end-user must respond to 10 questions because on the Google Form because It is a straightforward scale to administer to participants. The developer can use it on small sample sizes with reliable results. It can effectively differentiate between usable and unusable systems (Assistant Secretary for Public Affairs, 2013). The question for this system on figure 6.33. Figures 6.30, 6.31, and 6.32 provide information regarding the end-user and the oximeter's experience because experience is also one to determine this system is perfect for end-users, choosing this system is suitable for end-users with no experience.

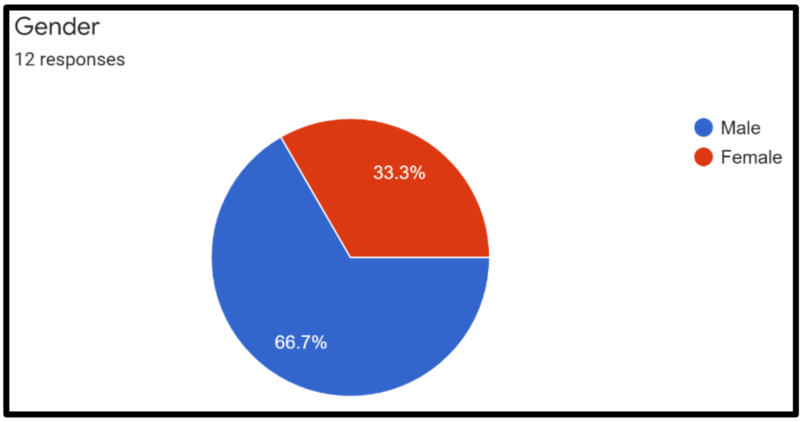


Figure 6.30: End-User Gender

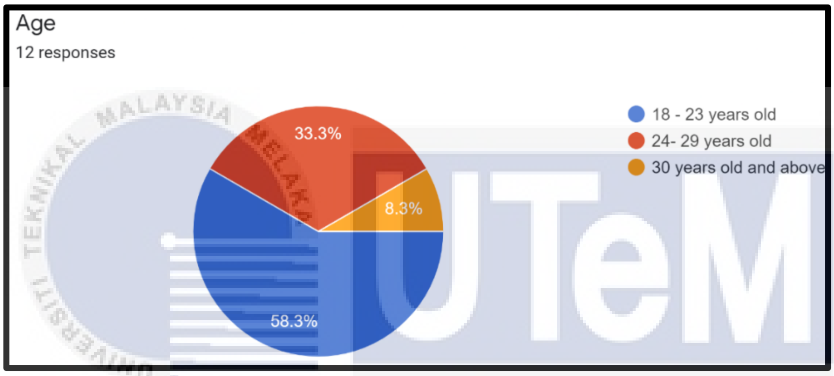


Figure 6.31: End-User Age

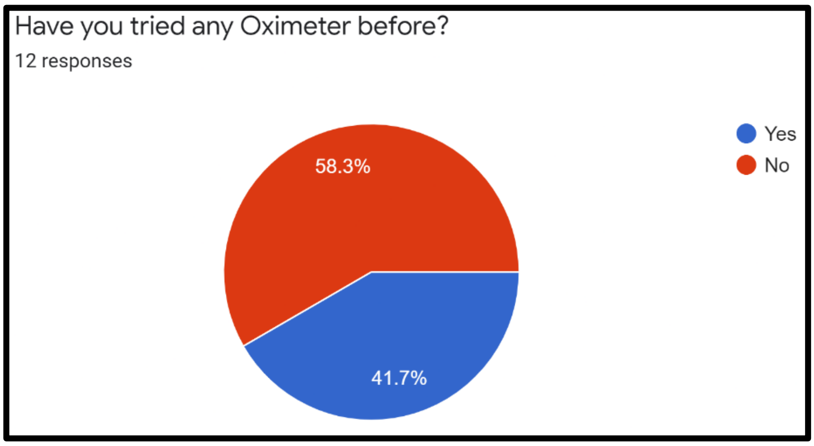


Figure 6.32: End-User Experience

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

Figure 6.33: Set of Questions (Assistant Secretary for Public Affairs, 2013)

Five options are available for responding to a question. It is based on a scale with one representing strongly disagree, two representing disagree, three representing neutral, four representing agree, and five representing strongly agree. All responses from the twelve end-users will be compiled into a histogram chart for easier reference, as illustrated in Figure 6.34. In figure 6.34, the Y-axis on the graph is the users' answer from strongly disagree to disagree strongly. For the X-axis is the users have been chosen from the developer from the friends and family. Legend with different colors in the Figure 6.34 is a question from 1 to 10 that have been answered for the users.

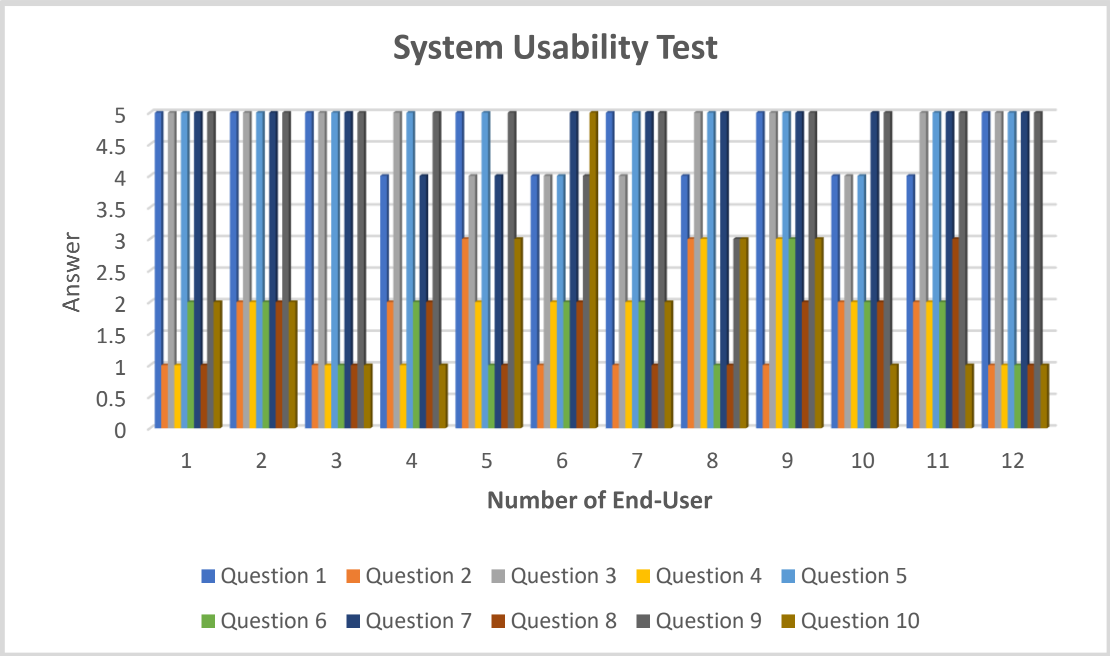


Figure 6.34: Bar Chart of System Usability Scale

Based on the end-user collection that shows in figure 6.34, the calculation score for System Usability Scale (SUS) is counted. The formula used to calculate the data is shown as follows (Nathan Thomas, 2019):

1) For each of the odd-numbered questions, need to subtract by one from the score given by end-users.
2) For each of the even-numbered questions, need to subtract by five from the score given by end-users.
3) Take these new values which you have found and add up the total score. Then multiply it by 2.5.

Table 6.14 summarizes the end-response users to each question. Twelve end-users were consulted to answer this System Usability Scale (SUS) question. After obtaining the end-response, user's the SUS must be calculated using the formula in Table 6.14. The total score is 88.42, the Development of DIY Oximeter for Health Monitoring is excellent or an A grade because of the system (Karampelas, Y., 2017). So, if the system gets a SUS score over 80, there is a good chance that the site will be recommended, while if it has a SUS score under 70, it may get negative reviews or comments.

Table 6.14: Table System Usability Scale (SUS)

Number of responses	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	SUS Raw Score	SUS Final Score
1	4	4	4	4	4	3	4	4	4	3	38	95
2	4	3	4	3	4	3	4	3	4	3	35	88
3	4	4	4	4	4	4	4	4	4	4	40	100
4	3	3	4	4	4	3	3	3	4	4	35	88
5	4	2	3	3	4	4	3	4	4	2	35	88
6	3	4	3	3	3	3	4	3	3	4	33	83
7	4	4	3	3	4	3	4	4	4	3	36	90
8	3	2	4	2	4	4	4	4	2	2	31	78
9	4	4	4	2	4	2	4	3	4	2	33	83
10	3	3	3	3	3	3	4	3	4	4	33	83
11	3	3	4	3	4	3	4	2	4	4	34	85
12	4	4	4	4	4	4	4	4	4	4	40	100
Average:												**88.42/100**

6.7 Conclusion

In conclusion, had achieved this project's objectives: to develop a prototype of the Development of a DIY Oximeter for Health Monitoring system, display the result using a smartphone, create an interactive alert notification, and test the effectiveness of the prototype. In the next chapter, we will cover the conclusion of the full project development.

CHAPTER 7: PROJECT CONCLUSION

7.1 Introduction

This chapter will summarize the project, from its inception through the testing phase. The project's goals are met, and this chapter discusses the project's contribution and constraints. This chapter is critical to the project's efficiency and improvement.

7.2 Project Summarization

This project utilizes the Internet of Things (IoT) with the NodeMCU and MAX 30100 sensor to build a warning system for heart rate and oxygen saturation when a threshold is surpassed. This project is concerned with monitoring heart rate and oxygen saturation and alerting users if the heart rate or oxygen saturation thresholds are exceeded using an algorithm. This system is designed to enhance the user's or a person's health status under any circumstance.

The first objective of this project is to develop low-cost hardware that can provide accurate readings. This objective has been achieved by finding a suitable microcontroller and sensor in this project. They have tested to determine the accuracy of this project almost the same with brand one and brand two and the prototype is also affordable and cheaper compared with brand one and brand two while the quality of measurement may be comparable. It is challenging to make sure the MAX 30100 sensor is suitable because the manufacturer already upgrades it.

Next, the second objective is to develop a software application that can display and record people oxygen saturation and heart rate. This objective has been achieved by developing an application using the Blynk software to display real-time monitoring and record the data from the accurate- time monitoring in the Blynk application.

Besides that, the third objective is to develop that can provide a user alert. The Blynk application achieves the objective, and the NodeMCU microcontroller allows a device connected to the smartphone. Whenever a heart rate is high, and oxygen saturation is low detected, an alert notification is sent to the Blynk application.

Furthermore, this project has achieved all the project objectives listed in Chapter I. This project is also user friendly, as stated in chapter VI, which is user testing. Users can view all the data only using a smartphone. Besides, it has a graph view, and the user can monitor the graph of the heart rate and oxygen saturation using an interactive graph, and the user also can download all the data in the CSV format. In conclusion, this project will provide safe health monitoring to the user and increase user awareness.

Finally, this project has a weakness because the prototype does not have a temperature to measure a human body temperature whether the user is in fever. The prototype also used direct electric current not from the rechargeable battery, such as the Li-ion battery. The user heart rate and oxygen saturation maybe not be the correct value compared to the certificate product. The prototype size of the device is 8.8 cm in length, 6.3 cm in breadth and 2.5 cm in height which is quite bulky.

Table 7.1: Strength and Weakness Prototype

Strength	Weakness
• Visualization of Data. • Alert Notification. • Record Data. • Increase user awareness. • Low-cost prototype.	• Temperature sensor. • Power source. • Accuracy of sensor. • Size quite bulky.

7.3 Project Contribution

This prototype is intended to assist humans in doing everyday tasks since heart rate and oxygen saturation are often affected by physiological variables such as the circadian cycle, posture, blood pressure, and physical exercise. It is a superior option since it saves time and alerts the user for what action must be done to avoid a worse outcome. Everyone immediately and universally uses this product. Individuals can easily monitor their heart rate and oxygen saturation (easy, inexpensive, practical).

7.4 Project Limitation

There are several limitations to this project. The limitation to this project is that the temperature sensor is not including in this project. The temperature sensor is also essential in health monitoring in the Covid-19 pandemic because some oximeter has only measured the heart rate and oxygen saturation but not the temperature for the user.

The other limitation of this project is the power source used to turn on when testing and the user to use this project or NodeMCU. The current power source is using direct electricity to turn on this. It may waste energy consumption because the voltage used is high. It is also troublesome when there is no electricity because there is no backup electricity like the Li-ion battery, AAA battery.

Another limitation to this project is that the MAX 30100 sensor is not too accurate for the heart rate reading, and it may have some differences a little from the exact user's or using brands 1 and 2. Chapter V compares the data with brand one and brand 2 to measure user heart rate and oxygen saturation. They have a slight bit error on the reading for this project compared to the exact heart rate that shows in brand one and brand two, but oxygen saturation has a very comparable value with brands 1 and 2. The MAX 30100 have a different version, such as MAX 30102, 30105 to measures the heart rate and oxygen saturation. Another version, MAX 3010X, maybe good to measures heart rate and oxygen saturation.

Next, need to buy commercial Blynk apps and design provides more data storage, MQTT because cannot export large data cannot export extensive data from Blynk in the Blynk local server database, and data is hard to find. The MQTT configuration is missing. Heart rate and oxygen saturation reading auto-delete after a specific reading.

Finally, this project's constraint is that the gadget must be lower in size. The gadget is 8.8 centimeters in length, 6.3 centimeters in width, and 2.5 centimeters in height, making it very large. Due to the breadboard used to create brand one or brand two, it becomes more difficult to 3D design.

7.5 Future Works

Future work will be required to overcome all the project's constraints, as mentioned in Chapter VII, to improve the project's function and layout. The project may be enhanced with the addition of a temperature sensor to monitor the human body temperature. The temperature important is some warning when the temperature exceeds the Health Ministry's specified limits during the Covid-19 pandemic.

Next, this project may be enhanced by including an independent power source powered by a rechargeable Li-Ion or AAA battery since both the NodeMCU and the MAX30100 sensor have the potential to operate at low current consumption when properly optimized and set. When compared to direct power, this stand-alone battery may significantly cut electric energy usage.

Furthermore, this idea may be enhanced by using a more precise sensor to get the data. This reading may aid in the data collection process and provide the best outcomes. For instance, the sensor can detect heart rate and oxygen saturation on anybody area and does it quickly.

Besides that, it can do future works to the system to make it more efficient comprehensive to storage data by providing it with an MYSQL or other platforms such as Thingspeak or cloud database. The platform most popular open-source database management system allows managing relational databases. The data can also be stored and exported using this platform with access to previous or real-time monitoring.

Finally, this method may be enhanced by decreasing the device's size. This prototype requirement will enhance comfort and contribute to a better ergonomic design. Perhaps this enhancement will improve readings due to its ability to connect to the wrist and fingers securely.

7.6 Conclusion

This project develops a simple DIY Oximeter for Health Monitoring to measure heart rate and oxygen saturation at minimum cost. The project is powered by a NodeMCU microcontroller attached with MAX 30100 to measure heart rate and oxygen saturation. This system not only provides the primary function of measuring human heart rate and oxygen saturation, but it does provide a notification to the users if the heart rate and oxygen saturation exceed the limits that have been set. The project met all the objectives stated in Chapter I, and the implementation and testing part is clearly shown in the previous chapter. Lastly, this project is a success and newly introduced to the current market. The Development of a DIY Oximeter for Health Monitoring covers the weaknesses of the current health monitoring system, and it performs more comprehensively and meets the user's requirements.

REFERENCES

1. Petersen, C. L., Chen, T. P., Ansermino, J. M., & Dumont, G. A. (2013). Design and evaluation of a low-cost smartphone pulse oximeter. Sensors, 13(12), 16882-16893.
2. Amien, D. M. B. M. (2018). Pulse Oximeter by arduino with GSM (Doctoral dissertation, University of Science and Technology).
3. Bakhri, S., Rosiana, E., & Saputra, R. C. (2020, March). Design of Low-Cost Pulse Oximetry Based on Raspberry Pi. In Journal of Physics: Conference Series (Vol. 1501, No. 1, p. 012003). IOP Publishing.
4. A. Dhadge and G. Tilekar, "Severity Monitoring Device for COVID-19 Positive Patients," 2020 3rd International Conference on Control and Robots (ICCR), 2020, pp. 25-29, doi: 10.1109/ICCR51572.2020.9344386.
5. Metcalfe, B., Iravani, P., Graham-Harper-Cater, J., Bowman, R., Stirling, J., & Wilson, P. (2021). A costeffective pulse oximeter designed in response to the COVID-19 pandemic. Journal of Open Hardware, 5(1).
6. Bhuyan, M. H., & Sarder, M. R. (2021). Design, Simulation, and Implementation of a Digital Pulse Oxygen Saturation Measurement System Using the Arduino Microcontroller. International Journal of Biomedical and Biological Engineering, 15(2), 105-111.
7. Miron-alexe, V. (2020). IoT PULSE OXIMETRY STATUS MONITORING FOR HOME QUARANTINED COVID-19 PATIENTS. Journal of Science and Arts, 20(4), 1029-1036.
8. Sihombing, P., Barus, Y. E., Sembiring, S., & Zamzami, E. M. (2020, June). The Development of Heart Rate Detection Using Arduino Microcontroller and Android. In *Journal of Physics: Conference Series* (Vol. 1566, No. 1, p. 012027). IOP Publishing.
9. N.V., R.K., M., A., E., B., J., S.J.P., A., K. and S., P. (2020), "Detection and monitoring of the asymptotic COVID-19 patients using IoT devices and sensors", International Journal of Pervasive Computing and Communications, Vol. ahead-of-print No. ahead-of-print. https://doi.org/10.1108/IJPCC-08-2020-0107
10. M. K. Altayeb, A. Abdelrahman, M. A. Bashir and O. A. Bashir, "Cost-effective Design of Pulse Oximeter using a Recycled SPO2 Sensor and Arduino Microcontroller," *2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, 2021, pp. 1-6, doi: 10.1109/ICCCEEE49695.2021.9429681.

11. Assistant Secretary for Public Affairs. (2013, September 6). *System usability scale (SUS)*. Usability.gov. https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html
12. NATHAN THOMAS. (2019, September 13). *How to use the system usability scale (SUS) to evaluate the usability of your website*. Usability Geek. https://usabilitygeek.com/how-to-use-the-system-usability-scale-sus-to-evaluate-the-usability-of-your-website/
13. Karampelas, Y. (2017, December 8). *What is SUS (System usability scale)?* Userfeel.com. https://www.userfeel.com/blog/sus

APPENDIX

Appendix A (Product Specification)

Brand 1 Speciation

Huawei Honor Band 5

(https://www.hihonor.com/global/products/wearables/honorband5/)

Brand 1	Specification
Honor Band 5	➢ Large AMOLED Touch Display. ➢ A Watch Face for Every Occasion. ➢ Continuous Heart Rate Monitoring. ➢ SpO2 Monitor for Everyone. ➢ Running schedule with fitness monitoring and customized suggestions shows statistics such as heart rate, exercise duration and distance, and calorie burn and aerobic/anaerobic results for a more detailed workout plan. ➢ This band is 50M water-resistant and swim proof. A six-axis sensor records speed, distance, calories, and SWOLF score for freestyle, backstroke, breaststroke, and butterfly. ➢ The HONOR Band 5 uses HUAWEI's TrueSleep technology to analyze sleep quality, detect sleep patterns, and offer over 200 customized sleep assessment recommendations.

	➢ Have you lost your phone? Fear not. The phone finder quickly locates it. Raise your wrist to preview messages and reminders, or snap selfies and group pictures. ➢ The AMOLED color display may be left on standby for up to 14 days. This band use the heart rate and sleep monitoring functions for up to 6 days. Battery life varies depending on use. ➢ Price: RM 127

Brand 2 Specification

Oxitech Pulse Oximeter

(https://shopee.com.my/product/231410908/8161262544?smtt=0.513328118-1629905278.3)

Malaysia Device Authority (MDA) Number

Brand 2	Specification
Oxitech Pulse Oximeter	➢ A small, noninvasive, and painless electronic device that is clipped to your fingertip. ➢ It is used to detect pulse rate (PR) and measure oxygen saturation level (SPO2) or the oxygen levels in the blood. It is one important tool for patients and health care providers to use to report back on overall disease management especially those with lung and heart diseases (e.g., COPD, asthma, pneumonia, heart failure, etc) and during the COVID-19 pandemic, it served as a potential tool to identify hypoxia (low blood oxygen saturation) which is one troubling sign of severe illness caused by COVID-19. ➢ Oxitech Pulse Oximeter Certified by Europe CE0123 ➢ Approved by Medical Device Authority (MDA) Malaysia (Reg No: GC1574221-53413) ➢ Dual colour OLED display ➢ Power saving ➢ User friendly Portable & convenient ➢ Battery Type: two AAA alkaline batteries ➢ 1 year warranty ➢ Price: RM 109

Appendix B (User Details)

➢ User 1 Details	
Age(y)	➢ 22
Gender	➢ Male
Skin	➢ Fair Skin
Weight (kg)	➢ 59
Height(m)	➢ 1.76

➢ User 2 Details	
Age(y)	➢ 17
Gender	➢ Male
Skin	➢ Fair Skin
Weight (kg)	➢ 50
Height(m)	➢ 1.69

➢ User 3 Details	
Age(y)	➢ 61
Gender	➢ Male
Skin	➢ Brown Skin
Weight (kg)	➢ 80
Height(m)	➢ 1.73

Appendix C (Day 1 to 5 Accuracy and Comparative Data for Heart Measurement (BPM))

DAY 1					
Brand	**User**	**Morning (BPM)**	**Evening (BPM)**	**Night (BPM)**	**Average / 3 Times**
1	1	79	79	89	**82.3**
	2	82	93	89	**88**
	3	84	99	91	**92.6**
2	1	81	78	95	**84.6**
	2	84	99	89	**90.6**
	3	80	91	92	**87.6**
3	1	81	76	88	**81.6**
	2	84	63	98	**81.6**
	3	79	91	89	**86.3**
DAY 2					
Brand	**User**	**Morning (BPM)**	**Evening (BPM)**	**Night (BPM)**	**Average / 3 Times**
1	1	99	86	78	**87.6**
	2	72	93	76	**80.3**
	3	96	89	98	**94.3**
2	1	100	85	77	**87.3**
	2	97	90	84	**90.3**
	3	96	90	99	**95**
3	1	97	85	76	**86**
	2	98	85	74	**85.6**
	3	95	90	99	**94.6**
DAY 3					
Brand	**User**	**Morning (BPM)**	**Evening (BPM)**	**Night (BPM)**	**Average / 3 Times**
1	1	84	82	65	**77**
	2	68	91	74	**77.6**
	3	115	108	91	**104.6**

2	1	75	84	76	**78.3**
	2	69	91	77	**79**
	3	115	109	93	**105.6**
3	1	78	83	60	**73.6**
	2	74	93	65	**77.3**
	3	108	106	93	**102.3**
			DAY 4		
Brand	**User**	**Morning (BPM)**	**Evening (BPM)**	**Night (BPM)**	**Average / 3 Times**
1	1	67	82	72	**73.6**
	2	79	96	70	**81.6**
	3	89	100	94	**94.3**
2	1	68	72	72	**70.6**
	2	81	98	73	**84**
	3	90	103	96	**96.3**
3	1	77	74	76	**75.6**
	2	75	89	75	**79.6**
	3	89	104	94	**95.6**
			DAY 5		
Brand	**User**	**Morning (BPM)**	**Evening (BPM)**	**Night (BPM)**	**Average / 3 Times**
1	1	78	68	67	**71**
	2	84	79	82	**81.6**
	3	84	86	85	**85**
2	1	76	67	70	**71**
	2	83	78	87	**82.6**
	3	92	91	91	**91.3**
3	1	80	67	73	**73.3**
	2	74	74	75	**74.3**
	3	89	86	84	**86.3**

Appendix D (Day 1 to 5 Accuracy and Comparative Data for Oxygen Saturation (SpO2))

DAY 1					
Brand	**User**	**Morning (SpO2)**	**Evening (SpO2)**	**Night (SpO2)**	**Average**
1	1	99	98	99	**98.6**
	2	97	99	98	**98**
	3	97	97	93	**95.6**
2	1	99	100	99	**99.3**
	2	100	100	99	**99.6**
	3	98	99	96	**97.6**
3	1	100	100	99	**99.6**
	2	96	97	98	**97**
	3	99	96	98	**97.6**
DAY 2					
Brand	**User**	**Morning (SpO2)**	**Evening (SpO2)**	**Night (SpO2)**	**Average**
1	1	96	99	96	**97**
	2	97	97	97	**97**
	3	99	94	97	**96.6**
2	1	99	99	99	**99**
	2	97	98	99	**98**
	3	98	98	96	**97.6**
3	1	99	99	99	**99**
	2	98	99	99	**98.6**
	3	99	99	95	**97.6**
DAY 3					
Brand	**User**	**Morning (SpO2)**	**Evening (SpO2)**	**Night (SpO2)**	**Average**
1	1	97	99	97	**97.6**
	2	97	99	98	**98**
	3	98	99	96	**97.6**

2	1	99	99	100	**99.3**
	2	100	98	99	**99**
	3	99	100	98	**99**
3	1	99	99	100	**99.3**
	2	100	98	100	**99.3**
	3	96	99	99	**98**
DAY 4					
Brand	**User**	**Morning (SpO2)**	**Evening (SpO2)**	**Night (SpO2)**	**Average**
1	1	97	99	98	**98**
	2	96	98	98	**97.3**
	3	96	96	96	**96**
2	1	99	99	99	**99**
	2	99	99	99	**99**
	3	99	98	96	**97.6**
3	1	100	100	99	**99.6**
	2	99	98	99	**98.6**
	3	99	96	95	**96.6**
DAY 5					
Brand	**User**	**Morning (SpO2)**	**Evening (SpO2)**	**Night (SpO2)**	**Average**
1	1	97	97	99	**97.6**
	2	99	99	95	**97.6**
	3	94	100	97	**97**
2	1	99	99	99	**99**
	2	99	99	99	**99**
	3	99	98	97	**98**
3	1	100	100	100	**100**
	2	97	99	98	**98**
	3	99	99	98	**98.6**