

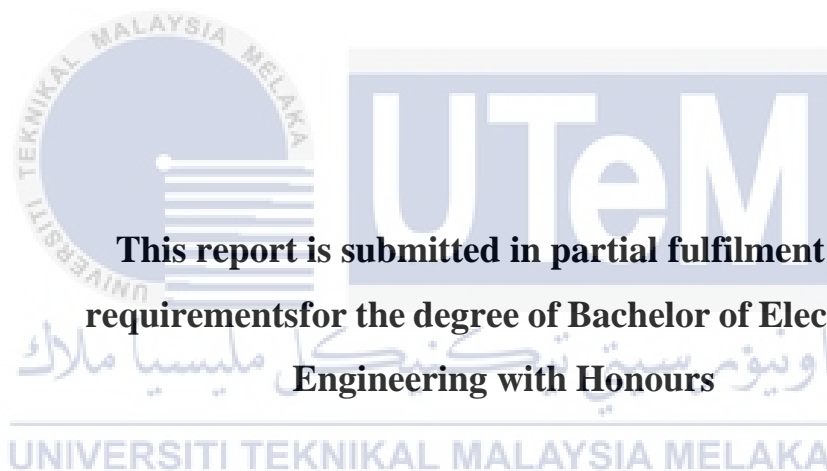
IOT BASED DEVICE FOR SELF-MONITORING OF COVID-19 PATIENT



UNIVERSITI TEKNIKAL MALAYSIA MELAKA

IOT BASED DEVICE FOR SELF-MONITORING OF COVID-19 PATIENT

NURUL IZZATI BINTI AYOP



**This report is submitted in partial fulfilment of the
requirements for the degree of Bachelor of Electronic
Engineering with Honours**

**Faculty of Electronic and Computer Engineering
Universiti Teknikal Malaysia Melaka**

2023

DECLARATION

I declare that this report entitled “**IOT BASED DEVICE FOR SELF-MONITORING OF COVID-19 PATIENT**” is the result of my own work except for quotes as cited in the references.



Signature :

Author : NURUL IZZATI BINTI AYOP
.....

Date : 19 JANUARY 2023
.....

APPROVAL

I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in terms of scope and quality for the award of Bachelor of Electronic Engineering with Honours.

Signature :

Supervisor Name : DR ANIS SUHAILA BINTI MOHD ZAIN

Date : 19 JANUARY 2023

DEDICATION

I dedicate this thesis to my parents, who have always believed in me and my abilities. Their unwavering love and support throughout my education has been a constant source of motivation for me. Without their sacrifices and encouragement, this accomplishment would not have been possible. I will always be grateful for their guidance and for the sacrifices they have made for me. I also dedicate this thesis to my supervisor, DR Anis Suhaila Binti Mohd Zain, who has been an invaluable mentor and friend throughout my final year project journey. Her expertise and knowledge in the field has been essential to the success of this project. I would like to thank her for her patience, guidance and for always being available to answer my questions. Without her support and encouragement, this thesis would not have been possible. Finally, I would like to express my gratitude to my friends who have supported me throughout this final year project journey. Their encouragement, feedback and friendship have been invaluable. I will always be grateful for the role they have played in helping me reach this goal.

ABSTRACT

The current COVID-19 pandemic has made many impacts to the world. The rate of infection of the COVID-19 virus increased, leading to an increase in the number of positive cases of COVID-19 reaching thousands of patients. Due to the sudden increase in positive cases in the short term, the hospital has to limit the use of medical facilities to the “red zone”. Patients are required to do their own home examination, where they need to spend a dollar to buy two different devices to take health readings. In addition, the hospital has a shortage of doctors and nurses to conduct patient-centred examinations. Tracing the modern state of technology 4.0, medical technology was created to help address the number of patients in the hospital. This technology is called self-monitoring that can be done alone at home. The project's first objective was to introduce a device that included pulse rate sensors, oxygen levels and temperature sensors. The second objective of the project is to make it easier for the medical team to serve patients easily by using BOT Telegram as a medium of communication between the medical team and the patient.

ABSTRAK

Pandemik COVID-19 saat ini telah memberikan pelbagai impak kepada dunia. Kadar keberjangkitan virus COVID-19 meningkat menyebabkan penambahan bilangan kes positif COVID-19 mencecah ribuan pesakit. Disebabkan kenaikan kes positif secara mendadak dalam jangka masa pendek, Pihak hospital terpaksa menghadkan kegunaan fasiliti perubatan kepada pesakit yang berada dalam ruangan yang dikenali “red zone”. Para pesakit dikehendaki membuat pemeriksaan sendiri di rumah, dimana perlu mengeluarkan wang ringgit untuk membeli dua device yang berbeza untuk mengambil bacaan kesihatan. Selain itu, pihak hospital mengalami kekurangan doktor dan jururawat untuk melakukan pemeriksaan keatas pesakit. Menyelusuri arus kemodenan teknologi 4.0, teknologi perubatan dicipta bagi membantu menangani jumlah pesakit di hospital. Teknologi ini dipanggil self-monitoring yang boleh dilakukan secara sendirian di rumah. Objektif pertama projek ini memperkenalkan sebuah device yang merangkumi sensor kadar nadi, kadar oksigen serta sensor suhu. Obejktif kedua projek ini adalah memudahkan pihak pasukan perubatan melayani para pesakit secara mudah dengan menggunakan BOT Telegram sebagai medium komunikasi antara pihak pasukan perubatan dan pesakit.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to the following individuals for their invaluable support and assistance during the final year project journey.

First, I would like to thank my supervisor, Dr Anis Suhaila Binti Mohd Zain, for her guidance, encouragement and expertise throughout the entire process. Her invaluable feedback and support were instrumental in the successful completion of this project.

I would like to thank my friends and family for their love and support throughout this process. Their encouragement and understanding helped me to stay focused and motivated.

I extend my sincere appreciation to all of the individuals and institutions who have directly or indirectly contributed to the completion of this thesis.

TABLE OF CONTENTS

Declaration

Approval

Dedication

Abstract

i

Abstrak

ii

Acknowledgements

iii

Table of Contents

iv

List of Figures

vii

List of Tables

ix

CHAPTER 1 INTRODUCTION

1

1.1 Project Background

1

1.2 Problem Statement

5

1.3 Objective

6

1.4 Scope of Work

7

1.5 Thesis Outline

7

CHAPTER 2 BACKGROUND STUDY

8

CHAPTER 3 METHODOLOGY	22
3.1 Block Diagram	23
3.2 Flowchart	24
3.3 Hardware	26
3.4 Software	31
3.5 Schematic Diagram	33
3.6 Hardware Prototype	35
3.7 Flow process	37
3.7.1 QR Code	37
3.7.2 Telegram BOT	38
3.7.3 Registration Form	39
3.7.4 Approval Doctor	40
3.7.5 Pairing Device	41
3.7.6 Self-Monitoring MAX30102 & MLX90614	42
3.7.7 Data Update on Telegram and Website	43
3.7.8 Doctor Analysis the History Patient with Sensor Reading and Give a Health Status	44
3.7.9 Patient Receive Result on the Telegram	45
CHAPTER 4 RESULTS AND DISCUSSION	46
4.1 General Measurement	47
4.2 Result for MAX 30102 (Heart Rate and Pulse Rate Sensor)	48

4.3	Result for Oxygen Saturation by using MAX 30102	vi 50
4.4	Result for temperature by using MLX 90614 Temperature Sensor	52
4.5	Project Significance	54
4.6	Project Sustainable Development Goals	55
4.7	Comparison of IoT-Based Device for Self-monitoring of COVID-19 Patient	56
CHAPTER 5 CONCLUSION AND FUTURE WORKS		57
5.1	Conclusion	57
5.2	Future Work	58
REFERENCES		59
APPENDICES		63
	Source Code for MAX 30102 (Heart Rate and Pulse Oximetry Sensor)	63
	Source Code for MLX90614 (Temperature Sensor)	77
	Source Code Arduino ESP 32 to Telegram in PHP	85

LIST OF FIGURES

Figure 1: Block Diagram of the proposed system.....	19
Figure 2: Graphical form from IBM Watson Platform	20
Figure 3: Block Diagram.....	23
Figure 4: Flowchart.....	24
Figure 5: NODEMCU ESP 32	26
Figure 6: 90614 Infra-Red Thermometer	27
Figure 7: MAX 30102 Pulse Rate and Heart Rate Sensor	28
Figure 8: The correct position touches the MAX30102 sensor.	29
Figure 9: Oxygenated hemoglobin (HbO ₂) in the arterial blood	30
Figure 10: Absorption (HbO ₂) vs Wavelength (nm)	30
Figure 11: BOT Telegram Applications.....	31
Figure 12: Website Interface	32
Figure 13: Schematic Diagram.....	33
Figure 14: Front view of prototype	35
Figure 15: Side View of prototype.....	35
Figure 16: Rear view of prototype	35
Figure 17: QR CODE.....	37
Figure 18: Telegram BOT interface	38

Figure 19: Registration form.....	39
Figure 20: Approval from Doctor	40
Figure 21: Pair patient with registered device.....	41
Figure 22: Successful pair the device.....	41
Figure 23: MAX30102 Heart Rate and Pulse Oximetry	42
Figure 24: MLX 90614 Temperature Sensor	42
Figure 25: Sensor Reading on Telegram BOT.....	43
Figure 26: Sensor Reading on Website	43
Figure 27: Patient Information Interface.....	44
Figure 28: Patient receive result from medical officer.....	45
Figure 29: General Measurement.....	47
Figure 30: Comparison Column Chart between IoT Device and Market Device	48
Figure 31: Comparison Column Chart between IoT Device and Market Device	50
Figure 32: Comparison Column Chart between IoT Device and Market Device	52
Figure 33: Sustainable Development Goals	55

LIST OF TABLES

Table 1: Patient diagnoses with COVID-19.....	3
Table 2:Symptoms of COVID-19 with category	3
Table 3: Connection of MAX30102 with ESP 32	34
Table 4: Connection of MLX90614 with ESP 32.....	34
Table 5:Data of heart beat rate by using IoT device and Market Device.....	48
Table 6: Data of oxygen saturation by using IoT device and Market Device	50
Table 7: Data of Temperature Sensor by using MLX 90614.....	52

اونيورسيتي تېكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA



اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

CHAPTER 1

INTRODUCTION



1.1 Project Background

The COVID-19 pandemic is currently one of the largest in the world problems faced by medical institutions. From November 19, 2020, it said that the total number of people in the world has been confirmed. More than 56.4 million people are infected with SARS-COV-2, and the total number of deaths from coronavirus is 1.35 million people proving that the number of COVID-19 cases is increasing rapidly worldwide [1]. Independent and convenient, healthy living is the aim of any human being irrespective of age, gender, location, or health status. However, there are limitations due to age, illness, medication, hospitalization, epidemic, pandemic, and alternative circumstances. Health-watching systems have evolved to help the convenient healthy living, additional accessible communication between healthcare givers and patients for close monitoring, the measure of important health parameters, routine consultation,

and overall healthy living. Moreover, with the recent advances in info and communication technologies (ICT) through the adoption of the net of Things (IoT) technology, smart health monitoring, and support systems currently have the next fringe of development and acceptableness for increased healthy living [2].

COVID-19 patients experience a variety of symptoms, including fever, shortness of breath, a drop in oxygen saturation, a dry cough and diarrhea, vomiting, sore throat, headache, loss of taste and smell, and body pain as well as abnormal pulse rate. High fever, low oxygen saturation level, and abnormal pulse rate are all considered serious symptoms. Hypoxemia and hypoxia are caused by low oxygen saturation levels and shortness of breath, respectively. Patients with hypoxemia and pulse rate issues have a lower chance of survival. Sometimes patients fail to recognize hypoxemia and an increasing rate of pulse and they die as a result of not receiving proper treatment. As a result, COVID-19 patients must be kept up to date on their health status, particularly their body temperature, heart rate, and oxygen saturation (SpO₂) [1].

The health department advises that every citizen should undergo a health screening when they reach the age of 18. This is because an early examination can detect early signs of a health condition before experiencing any physical symptoms. During the COVID-19 pandemic, the health department advises that every citizen is required to carry out self-exams as often as possible. This is because it can help in reducing contagion among humans when patients can detect early signs of COVID-19.

The parameters taken during the COVID-19 examination are heart rate, oxygen level, and body temperature. Heart rate is the number of heart beats per minute, also known as pulse rate. Calculating the pulses can be used to measure the pulse rate by increasing the blood flow volume. Healthy people have a heart rate that ranges

between 60 and 100 beats per minute. Adult males typically have a restful heart rate of 70 bpm, while adult females have a restful heart rate of 75 bpm [3]. Particular care should be taken when self-monitoring vital parameters such as blood oxygen saturation (SpO₂), the abnormal values of which are warning signs of a possible COVID-19 infection [4]. When assessing a person for COVID-19, body temperature the most typical COVID-19 symptoms is a fever or a raised body temperature, which can be a sign that the body is battling an infection. Table 1 shows the symptoms that need to take care of by the patient during COVID-19. While table 2 shows the symptoms of COVID-19 with category with analysis that has been found in these few years.

Table 1: Patient diagnoses with COVID-19

Mild Disease	Moderate Disease	Severe Disease
<ul style="list-style-type: none"> Fever/ Symptoms of upper respiratory tract infection SpO₂ ≥ 97% on room air 	<ul style="list-style-type: none"> Symptoms of pneumonia with RR ≥ 24 minute SpO₂ ≤ 94% on room air 	<ul style="list-style-type: none"> Respiratory disease requiring mechanical ventilation. SpO₂ ≤ 90% on room air.

Table 2: Symptoms of COVID-19 with category

Category	Symptoms
1	Asymptomatic
2A	<ul style="list-style-type: none"> Sore throat/ cold Cough Loss of sense of taste Loss of sense of smell

	<ul style="list-style-type: none"> • No fever • No difficulty breathing • Diarrhea twice in 24 hours • Nausea and vomiting • Fatigue • Muscle aches and pains
2B	<ul style="list-style-type: none"> • Fever/ onset of fever for more than two days • Shortness of breath when exerting yourself • Chest pain • Loss of appetite • Fatigue while doing daily activities or after waking up • Requires walking assistance • Increasing severity of symptoms (e.g., persistent cough, nausea, diarrhea) • Decreased level of consciousness • The gradual decrease in urine output over 24 hours
3	<ul style="list-style-type: none"> • Some breathing issues
4	<ul style="list-style-type: none"> • Requires oxygen support
5	<ul style="list-style-type: none"> • Needs intubation/ ventilator

The Internet of Things (IoT) offers new opportunities for applications such as smart cities and smart healthcare. Currently, the main uses of IoT in healthcare are remote monitoring and real-time health systems. In a time of crisis, such as the COVID-19 pandemic in 2020, IoT systems can help manage and control situations without imposing drastic restrictions on individuals and industries [5].

The proposed device integrates a temperature sensor and a heart rate sensor. It is designed to monitor body temperature, pulse rate, and oxygen saturation. It utilizes Telegram as a communication platform between COVID-19 patients and medical staff or healthcare personnel that can access and update a database containing information on registered patients. Through this platform, medical staff can provide instructions and guidance to both COVID-19 and post-COVID-19 patients.

1.2 Problem Statement

The problem statement is a crucial element of any project or research study, as it helps to define the scope of the work and serves as a clear and focused guide for solving the problems at hand. By accurately identifying and describing the problem or issue, the context in which it occurs, and its impact on stakeholders, a well-written problem statement can provide a clear and concise foundation for the project or study.

The rapid increase in daily active cases of COVID-19 is causing widespread concern globally and Malaysia is no exception. With over 30,000 daily active cases, the country is facing significant challenges in managing the pandemic. Two issues have been identified that contribute to this situation which are the lack of affordable and multi-functional health devices and the shortage of medical personnel to assist all patients. These issues highlight the urgent need for solutions to effectively manage the COVID-19 pandemic in Malaysia.

One of the driving factors behind this project is the lack of affordable and multi-functional health devices on the market. Currently, individuals must purchase separate devices to measure body temperature, pulse rate, and oxygen saturation, which can be

expensive. As a result, patients must spend money to purchase multiple devices to track their daily health status. This inspired the proposal for a single device that can measure all these vital signs.

Another issue that this project aims to address is the shortage of medical personnel available to assist all patients. Hospitals and clinics frequently receive a large number of patients in a single day and the lack of staff to conduct health checks can lead to medical personnel working overtime, especially during the COVID-19 pandemic. This can increase the risk of infection during hospital or clinic visits, contributing to the rise in daily cases of COVID-19. This issue is particularly relevant in Malaysia as highlighted by several articles discussing the country's shortage of medical personnel.

1.3 Objective

Achieving the objectives of this project is crucial to its success and serves as a guide throughout the project to ensure the desired goals are met. The first objective is to design a prototype of an embedded device that can simultaneously measure body temperature, oxygen saturation, and pulse rate. The second objective is to create an IoT communication platform using a Telegram BOT for patient data stored in a database. Through this platform, doctors can also send messages to patients about their health status.

1.4 Scope of Work

For this project, we need to use some hardware and software. The hardware includes an ESP 32 microcontroller, an Infrared Temperature Sensor to measure body temperature, a Pulse Oximeter Sensor to check pulse rate and oxygen levels, and a mobile phone for the Telegram app. The software we will use is the Arduino IDE to code the ESP 32 microcontroller.

1.5 Thesis Outline

This report is organized into five parts, each of which discusses a specific aspect of the project.

Chapter 1 provides an overview of the project, including the project background, problem statement, objectives, and scopes of work. Chapter 2 reviews similar projects that have been undertaken including their objectives and outcomes. Chapter 3 outlines the methodology used in this project, including the hardware and software used in this project. Chapter 4 presents the results of the project, including data analysis of measurements such as body temperature, heart rate, and oxygen saturation. Chapter 5, summarizes the entire project and future work that can upgrade this project easier to use and give an advantage to the people.

CHAPTER 2

BACKGROUND STUDY



Revolution 4.0 is a technological revolution incorporating automation that presents new challenges to the national economy. Automation, the Internet of Things (IoT), analytics and big data, simulation, systems integration, robots, and the cloud are some of the technologies that will help the country progress further. This new technology has the potential to solve issues in every sector of the country, as well as to develop the country and transform the country's future in the world of work. This technology has several advantages, particularly in the health industry. The technology intended for the health industry is where this technology may assist and relieve doctors' burdens in managing patient examinations. During the COVID-19 isolation phase, this equipment may remotely monitor and examine patients. It can help physicians work quicker and avoid face-to-face interactions.

Mohammad Monirujjaman proposed a smart health monitoring system for COVID-19 patients. In this project, the main objective is to develop and implement an IoT-based smart health monitoring system for COVID-19 patients based on human body temperature, pulse, and SpO2. This project has a system that displays the measurement of human body temperature, oxygen saturation, and pulse rate through a mobile application. As a comparison point, IoT Based Device for Self-monitoring of COVID-19 takes a reading of heart rate, oxygen rate, and temperature update through the BOT Telegram that has been specifically created for self-monitoring for COVID-19 patients [1].

Olutosin Taiwo proposed smart healthcare support for remote patient monitoring during COVID-19 quarantine. In this project, they create or develop a system that can monitor patients' health status and receive the doctors' prescriptions while self-quarantining at home. The doctors can diagnose ailments using the data collected remotely from the patient. One significant contribution of this project is patients in self-quarantine, they can use the platform that creates in this project to send or update daily health symptoms via their mobile phones which are Android. In comparison point, IoT Based Devices for Self-monitoring of COVID-19 can use in both systems which are the Android and iPhone operating systems. Also, the website can be viewed in both operating systems which is Android and iPhone (IOS) [2].

Md Milon Islam the project that he and his team proposed is the development of a smart healthcare monitoring system in an IoT Environment. Their project is about a smart healthcare system in an IoT environment that can monitor a patient's basic health signs as well as the room condition where the patients are now in real-time. Their project has five sensors used for the data of a heartbeat sensor, body temperature sensor, room temperature sensor, CO sensor, and CO2 sensor. In this project, the main IoT learning tool is ESP 32. As part analysis, they tested various subjects of different ages in different conditions. In the test, they manually calculated the actual and observed values from the developed system for heartbeat, body temperature, and room temperature sensors. The room temperature sensor is only used to measure humidity in this case. As a result, the rate of success between the observed data and actual data is approximately greater than 95% for all cases of the developed healthcare system. Even though the tests are performed outside of the hospital, authentic medical personnel can view and track the data in real-time. As a comparison, this LR uses the web interface where all the data is uploaded directly to the web interface, while IoT Based Devices for Self-monitoring of COVID-19 use the Telegram app as the medium of communication that easier to read data and upload the data directly to the web server [3].

Gabriella Casalino proposed a mobile health for contactless self-monitoring of blood oxygen saturation. In their project, they use a technique called a combination of Signal Processing and Computer Vision techniques. They analyzed the effectiveness of the proposed method, and a series of measurement comparisons were performed, with pulse oximeter values serving as the baseline. As a part of the analysis, the study included 21 subjects who varied in age, gender, and lifestyle. Because collecting measurements, from sick people, was impossible due

to the limitations imposed during the COVID-19 emergency, all subjects in this preliminary work were healthy. For a fair comparison, each subject's blood oxygen saturation was measured using a pulse oximeter and our contactless solution running on a laptop equipped with a camera. The Microsoft Lifecam HD7 camera was used. This is a small camera (3.44'' length, 1.57'' width), with autofocus and an HD 1080p sensor that ensures high sharpness and quality of the captured image. An empirical study with various cameras revealed that autofocus and at least 30 frames per second are the bare minimums for accurate results [4].

Seyed Shahim Vedaiei proposed an IoT Based System for Automated Health Monitoring and Surveillance in Post-Pandemic Life. This project consists of three parts which are a lightweight and low-cost IoT node, a smartphone application (app), and fog-based Machine Learning (ML) tools for data analysis and diagnosis. The IoT node monitors health parameters such as body temperature, cough rate, respiratory rate, and blood oxygen saturation, and then updates the smartphone app to show the user's current health status. The app alerts the user to keep a physical distance of 2 m (or 6 ft), which is critical in controlling virus spread. Furthermore, a Fuzzy Mamdani system (running on the fog server) takes into account environmental risk and user health conditions to predict the risk of infection spreading in real-time. The virtual zone concept conveys environmental risk and provides updated information for various locations. For communication between the IoT node and the fog server, two scenarios are considered: 4G/5G/WiFi or LoRa, which can be chosen based on environmental constraints. For various event scenarios, the required energy usage and bandwidth (BW) are compared. The COVID-SAFE framework can help to reduce the risk of coronavirus exposure [5].

Ali I. Siam proposed a portable health monitoring system that uses Internet of Things (IoT) technology to measure and track key vital signs, including heart rate, blood oxygen saturation, and body temperature. The system processes and encrypts the collected data using the Advanced Encryption Standard (AES) algorithm and then sends it to the cloud via an ESP8266 unit, which handles processing, encryption, and connectivity to the cloud over Wi-Fi. Hardware uses in this project are MAX 30102 sensor, DS18B20 sensor, and ESP8266 NodeMCUWiFi Devkit. As a comparison point, IoT Based Devices for Self-monitoring of COVID-19 use MLX90614 as a temperature sensor. MLX90614 is an infrared temperature sensor, which means it can measure the temperature of an object or surface by detecting the infrared radiation emitted by the project. This shows that MLX 90614 allows the measurement of the temperature of an object without requiring physical contact with the object, making it suitable for applications where contactless temperature measurement is desired. DS18B20 is a contact temperature sensor, which that's means it measures the temperature of an object by making physical contact with the object [6].

Vani Yeri proposed a system that consists of web and mobile applications based on continuous wireless monitoring of patients. Their objective for this project is to implement a low-cost system that also can transmit emergency vital signals. They use a pulse sensor, temperature sensor, oxygen sensor, and ECG sensor as the main components in their projects. They also use the WIFI module and GSM module for their cloud storage to store the information of patients and mobile application to send message alertness. The data from the sensor took every 30 seconds to see the result [7].

Gokcen Basaranoglu is researching a comparison of peripheral capillary hemoglobin oxygen saturation (SpO_2) values among every finger of the two hands. Based on their research, the right middle finger and right thumb had the highest and second highest average oxygen saturation values, respectively when measured using pulse oximetry in right-hand dominant volunteers. These values were statistically significant when compared to the values obtained from the left middle finger, which had the lowest average oxygen saturation. It is suggested that the right middle finger and right thumb may provide the most accurate reflection of arterial oxygen saturation [8].

Neha Arya proposed the system IoT-based Smart Health Monitoring Band for COVID-19 Patients. This project used the real-time parameter that are being recorded and send to the doctors and guardians by using the android app. In this project, they used a sensor called Infineon DSP 310, whereby the function is to detect the fever, blood pressure, and oxygen level. After sensing it, the data will send to the Arduino Nano. The data that has been recorded will send to the doctor and guardians by using the android app. In between sending the real-time data to the android app, they used ESP8266 (WIFI module) as the medium to communicate with the smartphone. As a result, they get data for the oxygen level is normal, if the oxygen level is less than normal, the device that has been created will send the alarm message to their family members and guardians. As for exercises that take the data from wave motion when the patient has begun his/her exercise. The fever result is the sense of body temperature by DPS 310, the sensor is reliable and able to sense the small temperature change. If the band finds spikes

or drops in temperature it will alert automatically by messaging. They also have researched and analysis doing fall accidents, which is the fall accident is the riskiest problem, especially for old people. The health band and accelerometer integrated with 6 Low PAN devices, collects data from the patients in real-time. If the system detects falls an alert is activated and an emergency message will send to doctors and guardians for their proper [9].

Bennet Praba proposed an IoT Based Handheld Smart Health Monitoring System for COVID-19. They present a design and implementation of an IoT-based project that is capable of recording the user's vitals such as heart rate, SpO2, and temperature which are these three vitals are important indicators that need to measure for COVID-19. In their project, they used a robust microcontroller prototype that contains various sensors such as heart rate, SpO2, and temperature. When the data is collected, it is sent to the microcontroller Arduino UNO which is in charge of forwarding it to the connected WiFi module (NodeMCU). The NodeMCU receives data from the Arduino and uploads it to a private cloud server, which is handled and maintained in the background by multiple PHP scripts and stored in a database. The same can be accessed via a website dedicated to combating and containing the spread of COVID-19, as well as mentioning current statistics and ways to prevent the virus [10].

Mwaffaq Otoom proposed a project about a real-time COVID-19 detection and monitoring system. The proposed title is An IoT-based Framework for Early Identification and Monitoring of COVID-19 cases. This proposed framework consists of five components which are real-time symptom data collection using wearable devices, treatment and outcome records from quarantine/isolation

centers, a data analysis center that uses machine learning algorithms, healthcare physicians, and a cloud infrastructure. The objective of their projects is to reduce mortality rates through early detection, following up on recovered cases, and a better understanding of the disease. The overall IoT based Framework for Early Identification and Monitoring of Novel Human Coronaviruses is all the symptom data collection and uploading through mobile phones and the data will be stored in cloud infrastructure from that the health physicians and data analysis center will analyze the symptoms that have upload by the patient, if the patient needs to go to the quarantine or isolation center the health physicians will contact the suspected case and health care centers. They use a machine learning algorithm for quick identification of potential COVID-19 infections which is a Support Vector Machine, Neural Network, Naïve Bayes, K-Nearest neighbor, Decision Table, Decision Stump, OneR, and ZeroR. In a conclusion, except for the Decision Stump, OneR, and ZeroR algorithms, all of these algorithms achieved accuracies of more than 90%. The use of the five best algorithms would provide effective and precise identification of potential COVID-19 cases [11].

Massimo Mapelli proposed a project about the Feasibility of remote home monitoring with a T-shirt wearable device in post-recovery COVID-19 patients. In this project, they are focusing on the post-COVID-19 patient. They used a function of the wearable device which is L.I.F.E a technologically advanced T- shirt device composed of ink-based dry electrodes linked to standard 12 lead ECG monitoring, 5 respiratory strain sensors, and 1 accelerometer. They analyze the L.I.F.E advanced T-shirt with 17 COVID-19 patients enrolled at hospital discharge. They underwent 7-17 days of monitoring to see the result of the T-shirt's advanced technology. The patients also completed the 2 hours monitoring period

a day during the days of monitoring. In a conclusion, the result shows the L.I.F.E t-shirt device can collect a full set of cardiorespiratory parameters during the days of monitoring [12].

Radovan Stojanovic proposed a project by using a headset it can track COVID-19 symptoms. The design of this project is by using the simplest sensor consisting of any mobile phone with a standard headset with a built-in microphone. It connects to a microphone/speaker input (via a 3.5mm jack) and detects breathing issues, respiration rate, and cough. By using an audio signal which is they use MATLAB software to detect the respiration problem [13].

Md Safwan Mondal proposed and developed for remote temperature monitoring device which is experimentally proven to perform with 98% accuracy. Their project is about the design and development of a wearable remote temperature monitoring device for smart tracking of COVID-19 fever. The main in their hardware is NodeMCU, DS18B20 Temperature Sensor, Buzzer, and Vibrator. Before the data send to the server, they will have a vibration when using this wearable device that shows the device has been connected to the WiFi. The data will send to the ThingSpeak server. If the fever crosses the limit of fever temperature which is 38 degrees Celsius, the device will start giving an alert by turning the buzzer on so that the person acknowledges the temperature that he/she has. As a result, the DS18B20 needs to do the calibration during the use of that sensor. They tested the device during different times of the day, the temperature was tracked and compared with Vick's Thermometer's temperature value. They take a real-time measuring temperature and display it with a graph in ThingSpeak windows. As a conclusion from their project, their devices provide non-contact or

distant accessing the scope for body temperature monitoring which can play a pivotal role in the prognosis of COVID-19 and other contagious diseases. The accuracy of 98% between actual temperature values and device observed values with their calibrations [14].

Safaa N. Saud Al-Humairi in this project proposed a COVID-19 real-time system for tracking and identifying suspected cases using an Internet of Things platform for capturing user symptoms and notifying the authority. There are proposed framework addressed four components which are real-time symptom data collection via a thermal scanning algorithm, a facial recognition algorithm, a data analysis that uses an artificial intelligence algorithm, and a cloud infrastructure. A monitoring experiment was carried out to test three different ages, child, middle, and older, taking into account the scanning distance influence when compared to contact wearable sensors. The results show that 99.9 percent accuracy was achieved within a (500 ± 5) cm distance, with accuracy decreasing as the distance between the camera and the objects increased. The results also revealed that the scanning system's accuracy had changed slightly as the ambient temperature dropped below 27 °C. Based on the simulated high-temperature presence, the system demonstrated an effective and immediate response by sending an email and an MQTT message to the person in charge of providing accurate identification of potential COVID-19 cases [15].

Puput W. Rusimamto proposed to develop an effective and efficient thermogenic for early detection of COVID-19. They used Arduino Pro mini Atmega328p - 8 BIT AVR controller and the camera used is the Esp32 Cam WiFi + BT Soc Module. E18-D80NK IR Obstacle Avoidance Proximity with the

infrared light reflection to detect objects or obstacles at a certain distance. For the result, the average error is calculated to be 0.6% with a standard deviation of measurement using the thermogenic of 0.088 and a standard deviation of measurement using the research result of 0.078 [16].

Faisal Jamil in this project proposed a system that was designed and developed using Hyperledger fabric, an enterprise-distributed ledger framework for creating blockchain-based applications. This method offers several advantages to patients, including an extensive, immutable history log and global access to medical information from anywhere at any time. To collect the physiological data, the Libelium e-Health toolkit is used. Using a standard benchmark tool known as Hyperledger Caliper, the performance of the designed and developed system is evaluated in terms of transaction per second, transaction latency, and resource utilization. It is discovered that the proposed system outperforms the traditional healthcare system in terms of patient data monitoring [17].

C.Premalatha with his/her team proposed a project namely human health monitoring system. The main goal is for the specialist to monitor the patient's condition from a distance to save the patient's life. This system is expected to be reasonably priced. Embedded technology will be used to easily monitor the patient's condition. The primary goal of this project is to monitor the patient's body temperature and display it to the doctor. The hardware in this project used is Arduino Uno, LM35 Temperature Sensor, and Heartbeat sensor. The sensor's information received by the data is transferred to the cloud via the Arduino board. The data is continuously sent to the cloud, where doctors can view the patient details, Data on patient health parameters is stored in the cloud and the doctor can

view it. At the same time, the data from the sensor will send to the mobile phone by using a GSM modem. The data will send to the physician. In IoT Based Devices for Self-monitoring of COVID-19, use NodeMCU ESP 32 as a microcontroller that has built-in WiFi that can send data from the Telegram application to the website without having external WiFi. The MLX90614 has a higher level of accuracy than the LM 35 which has a low level of accuracy. The MLX90614 can measure without contact or contactless while the LM35 is a contact temperature sensor that needs physical contact [18].

Amandeep Kaur and team propose this system is designed to monitor vital body parameters such as pulse rate and body temperature using dedicated sensors and a Raspberry Pi connected to the internet through IoT. The system is wearable and also allows for remote health monitoring by storing collected data in the Bluemix cloud, where it can be accessed and analyzed by a doctor. To ensure an optimal balance of accuracy and cost, the system uses the DS18B20 temperature sensor and the KG011 heart rate sensor. The readings taken by these sensors are displayed in graphical form on the IBM Watson IoT platform. The figure 1 shows the block diagram of the proposed system. While the figure 2 is graphical form from IBM Watson platform. [19].

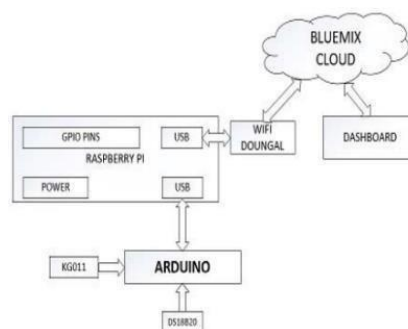


Figure 1: Block Diagram of the proposed system

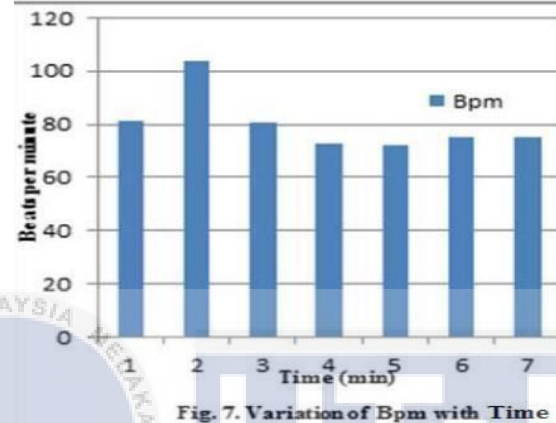
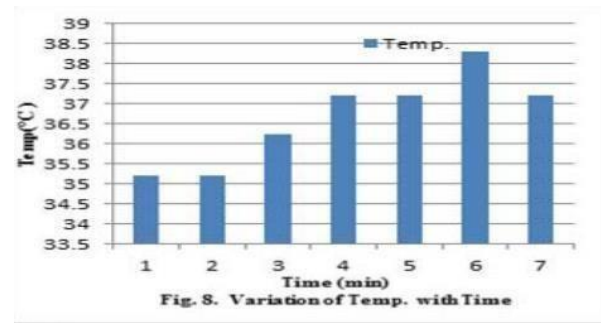


Figure 2: Graphical form from IBM Watson Platform

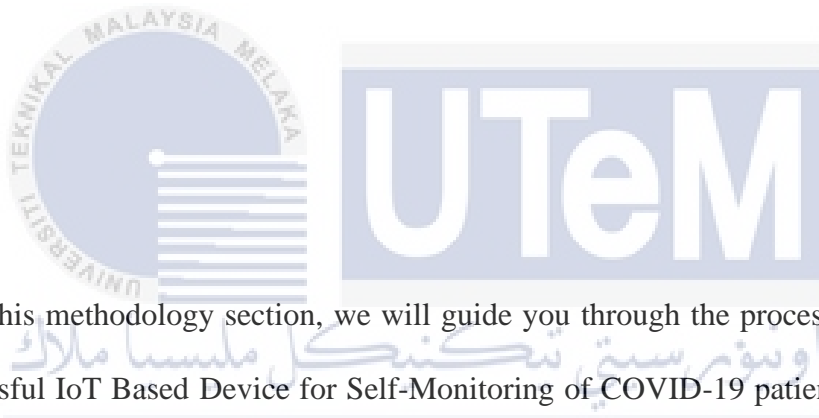
Sharanya Mahapatra and team propose an IoT system that has been developed to track and trace the movements of individuals using RFID and GPS technology. This system uses geofencing to set virtual boundaries and monitor the movements of infected individuals and also includes a fingerprint scanner for authentication and a real-time database for managing health status information. The system is designed to effectively control the spread of infectious diseases within organizations and workplaces and is effective in identifying infected individuals through validation testing. In a conclusion, the RFID and GPS-enabled wristband used in this system not only tracks the current location of the user but also records their entire travel path and contacts. The fingerprint scanner helps to authenticate individuals and prevent misuse of wristbands. The geofencing algorithm in the system not only sets boundaries for infected individuals but also monitors whether

they remain within those boundaries. If an infected individual move outside of the designated boundary, the system will generate an alarm. The system being proposed includes all necessary modules that operate in real-time, allowing for the communication of health status, travel path, and contact history to authorized individuals through email and a secure web portal [20].



CHAPTER 3

METHODOLOGY



In this methodology section, we will guide you through the process of creating a successful IoT Based Device for Self-Monitoring of COVID-19 patient project from start to finish. First, will introduce the block diagram, which outlines the various components and microcontrollers that are essential for the project. Then, walk through the flowchart, which outlines the process of building the project, starting with the hardware and culminating in the software. After that will delve into the specifics of the hardware and software that are used and explain the reasoning behind the hardware and software. Finally, will show the process of installing the hardware onto the microcontroller, and provides a schematic diagram for reference.

3.1 Block Diagram

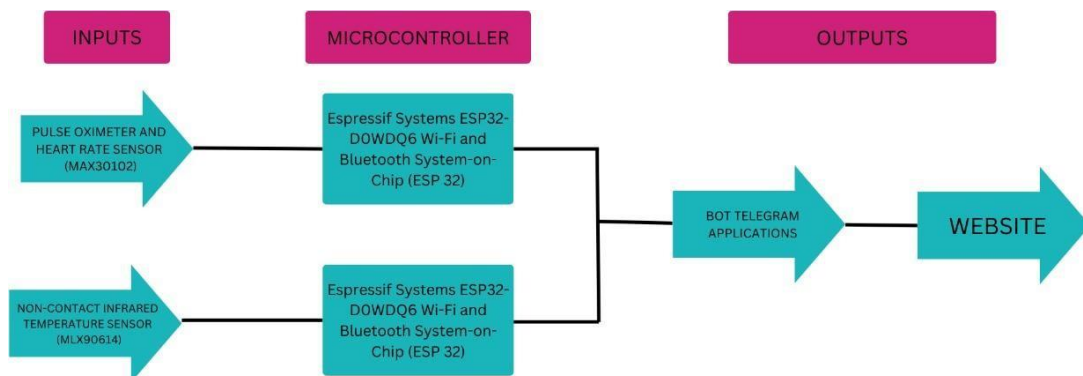


Figure 3: Block Diagram

As depicted in Figure 3, the block diagram for the proposed IoT Based Device for Self-monitoring of COVID-19 patients project includes two inputs sensor: a pulse oximeter and heart rate sensor (MAX30102) and a non-contact infrared temperature sensor (MLX90614). These sensors will be connected to an ESP 32 microcontroller, specifically the Espressif Systems ESP32-D0WDQ6 Wi-Fi and Bluetooth System-on-Chip. The data collected by these sensors will be displayed on a Telegram Bot application specifically designed for this device. Additionally, the data will be transmitted to a website that is only accessible to medical professionals, who will use it to determine whether the patient may be infected with COVID-19. If necessary, the hospital or doctor will use the Telegram BOT application to notify the patient that they need to self-quarantine or seek further examination at a nearby healthcare facility.

3.2 Flowchart

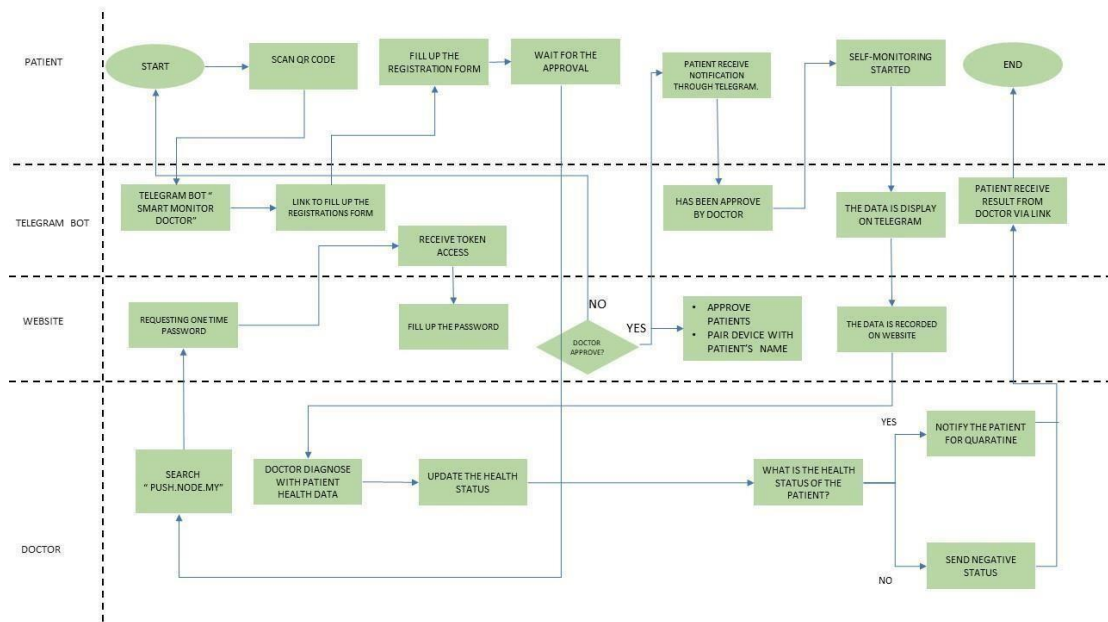


Figure 4: Flowchart

Based on figure 4 above, show the flow chart or process flow for the IoT-Based Device for Self-Monitoring of the COVID-19 patient project. Divided into four parts involved in IoT-Based Devices for Self-Monitoring of COVID-19 patients namely patient, website, Telegram BOT, and doctor.

The start of process begins with the patient, where the patient is asked to fill out a registration form after scanning the QR code provided with the device. After completing the registration, the patient is asked to click the submit button for approval from the medical officer. The patient's approval will be approved by the medical officer on a website called "push.node.my".

Medical officers visit the website "push.node.my" where patient registration will be approved on this website. Before the medical officer accesses his account, the website will ask for an access token or better known as a one-time password which will be given through a Telegram BOT reserved for medical officers. After filling in the access token obtained from the Telegram BOT, the medical officer will approve the patient's

registration after viewing the content of the registration. If the patient does not get approval from the medical officer, the patient needs to start from the beginning. If the patient gets approval from the medical officer, the patient will receive notification through Telegram BOT, that their registration has been approved by the medical officer. Patient registration information can be viewed through the provided information tab. The medical officers will check and pair it with a device registered with the Ministry of Health Malaysia according to the patient's name.

Patients will begin self-monitoring after receiving registration approval and the device to be used has been successfully paired. Among the parameters provided in the IoT-Based Device project for Self-Monitoring of COVID-19, patients in terms of measuring body temperature, measuring heart rate, and measuring oxygen levels. The data will be displayed on the registered Telegram BOT. It will also be updated directly to the website "push.node.my" where medical officers can see the patient's reading rate by name of each patient.

After the patient takes or measures body parameters, the readings will be updated directly on the website. Only medical officers with registered profile accounts can access this website. After the patient has finished measuring their body parameters, the medical officer will conduct a study or analysis based on the patient's profile to find out the patient's status either in the positive group of COVID-19 or negative group of COVID-19. If the patient is found to be positive for COVID-19, the medical officer will inform the patient to self-quarantine at home via Telegram BOT. If the patient is found to be negative for COVID-19, the medical officer will notify the result via Telegram BOT.

3.3 Hardware



Figure 5: NODEMCU ESP 32

Based on figure 5 above shows the NODEMCU ESP 32 that use in this project as a microcontroller. IoT Based Devices for Self-monitoring of COVID-19 patients choose the NODEMCU ESP32 because have features related to the internet. The NODEMCU ESP 32 has a single or dual-core 32-bit LX6 Microprocessor with a clock frequency of up to 240 MHz. This microcontroller has 520KB of SRAM, 448 KB of ROM, and 16 KB of RTC SRAM. It supports 802.11 b/g/n Wi-Fi connectivity with speeds up to 150 Mbps. This microcontroller has 34 programmable GPIOs and up to 18 channels of 12-bit SAR ADC and 2 channels of 8-bit DAC. The serial connectivity includes 4 x SPI, 2 X 2 x I²C, 2 x I²S, and 3 x UART. The ESP 32 has ethernet MAC for physical LAN Communication, 1 host controller for SD/SDIO/MMC, and 1 slave controller for SDIO/SPI. It also has motor PWM and up to 16 channels of LED PWM. The secure boot and the flash encryption. The cryptographic hardware acceleration for AES, Hash (SHA-2), RSA, ECC, and RNG.

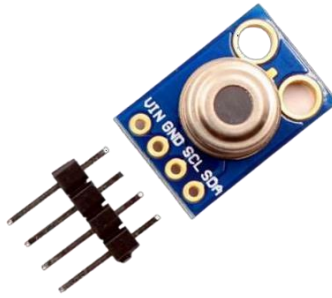


Figure 6: 90614 Infra-Red Thermometer

Figure 6 above shows MLX 90614 Infrared Thermometers that have been usually used to measure body temperature. The MLX90614 is a sensor that can measure the temperature of an object without coming into direct contact with it, as well as the temperature of the sensor's surroundings. The temperature measured by the sensor is referred to as the object temperature, while the temperature of the sensor's surroundings is referred to as the ambient temperature. Both temperatures can be determined by the MLX90614. As for this project, IoT Based Device for Self-monitoring of COVID-19 patients only measures object temperature. The size of this sensor is small and the cost for this sensor is low. The calibration for this sensor is $-40 - +125^{\circ}\text{C}$ sensor temperature and -70 to $+380^{\circ}\text{C}$ object temperature. The accuracy of this sensor is high in a wide temperature range. The versions for this sensor are single and dual. The voltage that is compatible used for this sensor is 3V. The MLX90614 is a special sensor that can measure temperature without touching the object. It has a filter inside that helps it ignore light that we can see or light that is very close to the light we can see. This helps the sensor give accurate temperature readings. The sensor is also able to work well even if there is a lot of light around it, like sunlight.

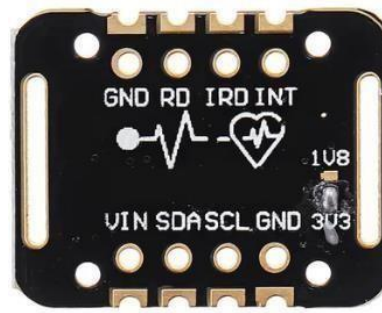


Figure 7:MAX 30102 Pulse Rate and Heart Rate Sensor

Figure 7 above shows the MAX30102 Pulse Oximeter and Heart Rate Sensor is an i2C-based low-power plug-and-play biometric sensor. The best voltages that can use for this sensor are 1.8V and 3.3V. The design of this sensor is an integrated LEDs, Photo sensor, and High-Performance Analog Front – End. The size of this sensor is a tiny 5.6mm x 2.8mm x 1.2 mm and 14 pins optically enhanced system in the package. The MAX30102 is a device that is used to measure pulse and heart rate using light. It consists of two high-intensity light-emitting diodes (LEDs), one emitting red light at 660 nanometers (nm) and the other emitting infrared (IR) light at 880 nm. It also has a photodetector, which is used to measure the amount of reflected light. When the LEDs are shone onto a part of the body, such as the finger or earlobe the photodetector measures the amount of light that is reflected. This method of measuring pulse and heart rate through the use of light is known as photoplethysmography.

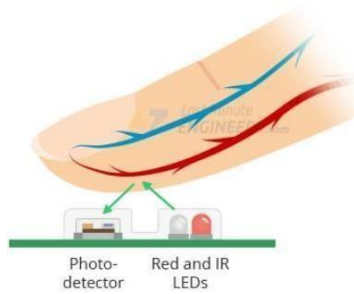


Figure 8: The correct position touches the MAX30102 sensor.

The MAX 30102 is a device that has two main functions which are heart rate measurement and pulse oximetry. Heart rate measurement involves using the device to determine the number of times that the heart beats per minute. Pulse oximetry involves using the device to measure the oxygen level in the blood. The MAX30102 can measure heart rate and oxygen saturation in the blood using infrared light. Oxygenated hemoglobin in arterial blood absorbs infrared light and the more oxygenated the blood is, the lighter it absorbs. The redder the blood (the higher the hemoglobin), the more IR light is absorbed. As the blood is pumped through the finger or other part of the body with each heartbeat, the amount of reflected light changes, creating a fluctuating waveform on the photodetector. By shining light and constantly measuring the reflected light with the photodetector, the device can quickly determine the heart rate. The changing waveform is used to calculate the number of heart beats per minute. This explanation is shown in figure 8 above.

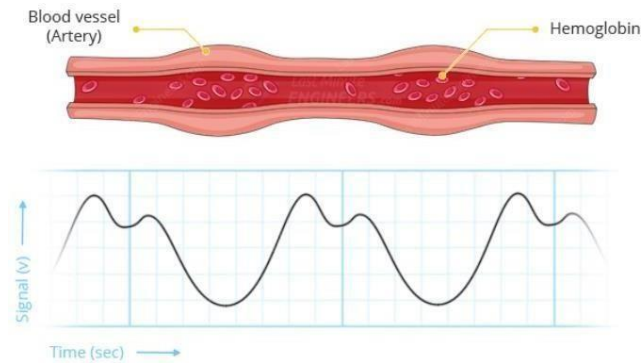


Figure 9: Oxygenated hemoglobin (HbO₂) in the arterial blood

Pulse oximetry is a method of measuring the oxygen saturation level in the blood using light. This technique is based on the fact that the amount of red and infrared light absorbed by the blood varies depending on the oxygen content of the blood. The absorption spectrum of oxygenated hemoglobin (HbO₂) and deoxygenated hemoglobin (Hb) is shown in the following graph. This spectrum illustrates how the absorption of light changes as the wavelength of the light changes and it is used to determine the oxygen saturation level of the blood. From the graph shown in figure 10, deoxygenated blood absorbs more RED light (660nm) while oxygenated blood absorbs more IR light (880nm).

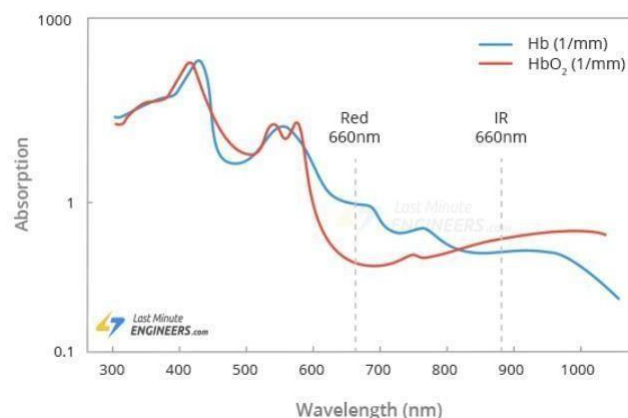


Figure 10: Absorption (HbO₂) vs Wavelength (nm)

3.4 Software

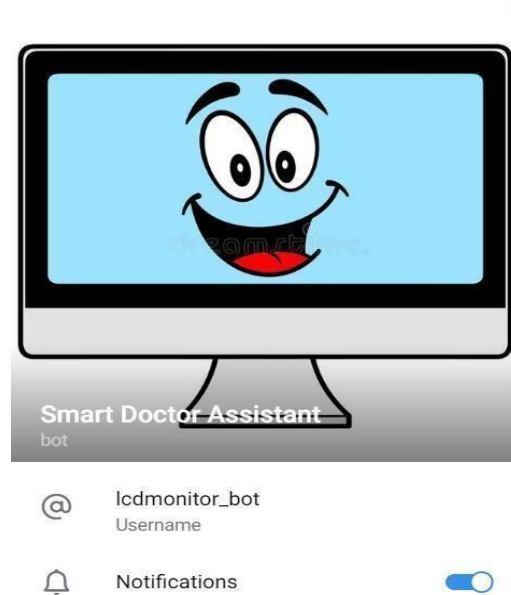


Figure 11: BOT Telegram Applications

The “Smart Doctor Assistant” is a Telegram BOT that is used to display the results of sensor readings, specifically the temperature sensor (MLX90614) and the heart rate and pulse rate sensor (MAX30102). This display which is shown in figure 11, is used to present the readings from these sensors to the user. In addition to displaying the sensor readings, the “Smart Doctor Assistant” Telegram BOT also indicates when the device is ready to take readings from both sensors. The device, which is connected to the Telegram BOT, is used to obtain the readings from the temperature and pulse sensors.

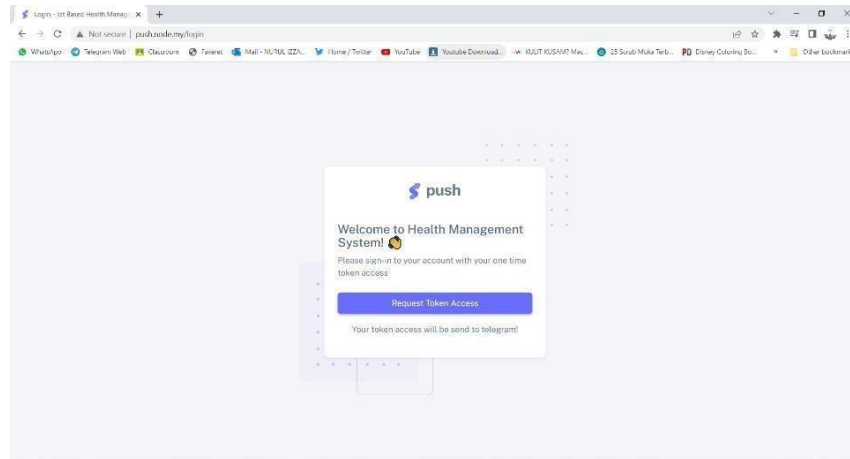


Figure 12: Website Interface

Figure 12 shows the interface of a website that is used to display various information about a patient, including their profile and history, the result of two sensors that are used to monitor the patient, and the information about the device that is registered to the website. This website can only be accessed and viewed by doctors or medical officers who have an account and are authorized to do so. The website displays the profile and history of the patient, as well as the results of the two sensors that are used to monitor the patient's health. It also includes information about the device that is registered on the websites. This information is intended to be used by medical professionals to monitor the health of the patient and to make any necessary medical decisions.

3.5 Schematic Diagram

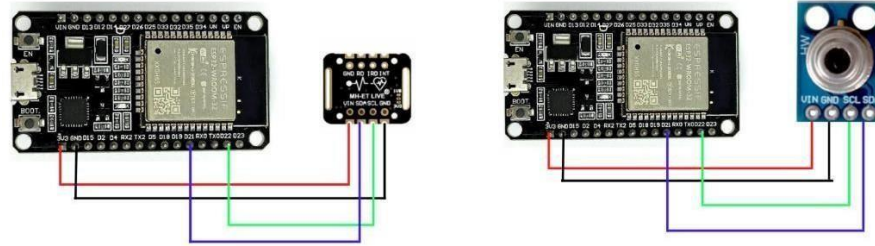


Figure 13: Schematic Diagram

Based on figure 13 shows the schematic diagram for this project. The figure shows the ESP 32 as a microcontroller, pulse rate and heart rate sensor (MAX30102), and temperature sensor (MLX90614). For the pulse rate and heart rate sensor (MAX30102) use all the pins. The Vin at heart rate sensor is connected to a 3.3V ESP32. The GND pin is connected to the GND ESP 32. The SCL pin is connected to GPIO 22. The SDA pin is connected to GPIO 21. For the temperature sensor (MLX90614) use all the pins. The Vin at temperature sensor is connected to 3.3V ESP32. The GND pin is connected to the GND ESP 32. The SCL pin is connected to GPIO 22. The SDA pin is connected to GPIO 21.

Table 3: Connection of MAX30102 with ESP 32

Heart Rate and Pulse Rate Sensor (MAX 30102)	MICROCONTROLLER ESP 32
Vin	3.3V
GND	GND
SCL	GPIO22
SDA	GPIO 21

Table 4: Connection of MLX90614 with ESP 32

Temperature Sensor (MLX90614)	MICROCONTROLLER ESP 32
Vin	3.3V
GND	GND
SCL	GPIO22
SDA	GPIO 21

3.6 Hardware Prototype

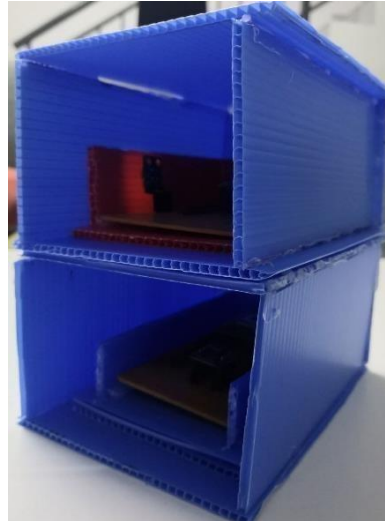


Figure 14: Front view of prototype



Figure 15: Side View of prototype

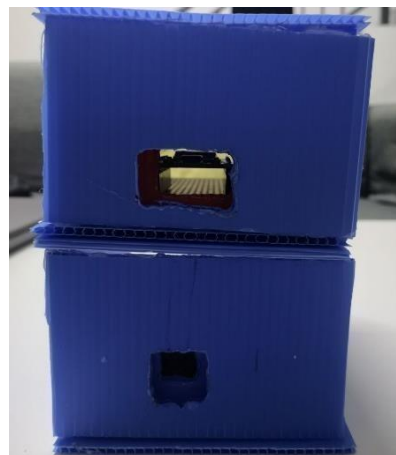


Figure 16: Rear view of prototype

The three diagrams provided are prototypes for an IoT Based Device for Self-monitoring of COVID-19 Patients. This prototype is made to give an idea of how this project will be saved or what form this project will take.

The prototype that is depicted in the first figure, figure 14 was crafted to be easily accessible and removable for troubleshooting purposes in case of any problems with the circuit. This prototype was constructed with two levels or floors, where the ground floor or first floor is designated for measuring pulse and oxygen saturations. While the second floor or upper floor is designated for taking body temperature readings. This layout makes it convenient for the patient to place their finger on the MAX30102 sensor, which is a Heart Rate and Pulse Oximetry sensor. Also, make it effortless for the patient to take a body temperature reading.

The diagram in figure 15 illustrates a side view of the prototype for this project. It features a temperature sensor known as the MLX 90614, which is designed to make it simple for patients to place the device on a flat surface for accurate measurement of body temperature.

The third diagram, figure 16, shows the rear view of the prototype, where the USB plug is located for supplying to the ESP 32 microcontroller, which in turn powers the MAX30102 Heart Rate and Pulse Oximetry Sensor, as well as the MLX 90614 Temperature Sensor.

3.7 Flow process

3.7.1 QR Code



Figure 17: QR CODE

Figure 17 shows a QR code that, when scanned, will open a Telegram BOT called “Smart Doctor Assistant”. This bot, which has been created by the health department, is specifically for device users and will allow them to register and view readings, the QR code has been provided to make it easier for the patient to access the bot. A QR code is provided to make it easier for patients to use this device. This QR code also makes it easier for patients to go directly to Telegram BOT without having to search for the name of Telegram BOT because this QR Code directly sends to Telegram BOT "Smart Doctor Assistant". This QR Code is made to make it easier for patients to get information or more precisely to contact medical officers to ask about symptoms faced by patients. This QR Code is made to make it easier for patients to get information or more precisely to contact medical officers to ask about symptoms faced by patients.

3.7.2 Telegram BOT

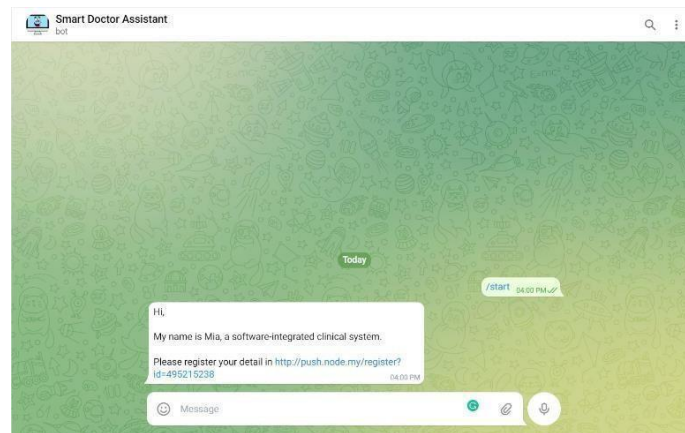


Figure 18: Telegram BOT interface

Figure 18 shows the interface of the Telegram BOT “Smart Doctor Assistant” which is displayed after the user or patient scans the provided QR code on registered devices only. This view will display a link for the patient to register their profile for hospital records. This Telegram BOT will record all the readings taken by the patient and provide notifications through the BOT. It will only be accessible to registered patients, who will each be given a unique Telegram ID to view their readings and hospital notifications.

3.7.3 Registration Form

Registration Form

Self Information

Name
Mohd Ali Bin Abu

Identity Card No.
9001000000000000

Age
0

Mobile No.
01234567890

Telegram ID
485215238

Address
No 10, Jalan Hang Tuah, 90022, Melaka, Malaysia

Health Information

Do you have diabetes?
☒ No
☐ Yes

Do you have high blood pressure?
☒ No
☐ Yes

Do you have asthma?
☒ No
☐ Yes

Have you had operation before?
☒ No
☐ Yes

Fever
☒ No
☐ Yes

Cough
☐ No
☐ Dry Cough
☐ Cough With Mucus
☐ Normal Cough

Shortness of breath
☒ No
☐ Yes

Fatigue
☒ No
☐ Yes

Body Aches
☒ No
☐ Yes

Loss of Smell / Taste
☒ No
☐ Yes

Sore Throats
☐ No
☒ Yes

Diarrhea
☐ No
☐ Yes

Nausea or Vomiting?
☐ No
☐ Yes

How many days have you been sick?
☒ Less than 7 days
☐ More than 7 days

Heart Diseases
☒ No
☐ Yes

Any other diseases? (Optional)

Figure 19: Registration form

The registration form provided by the hospital to patients is depicted in diagram 19. This registration form is for future hospital or clinic records as additional knowledge if the patient suffers from an illness that requires several specialist doctors to analyses or diagnoses. The hospital asked questions about COVID-19 symptoms on the registration form. Among the symptoms of COVID-19 that have been raised are fever, cough, shortness of breath, fatigue, body aches, loss of smell/taste, sore throats, diarrhea, and nausea or vomiting. In terms of other illnesses, does the patient have diabetes, high blood pressure, asthma, or heart disease, or has he ever undergone surgery. Furthermore, the registration form inquires as to how many days the patient has been ill.

3.7.4 Approval Doctor

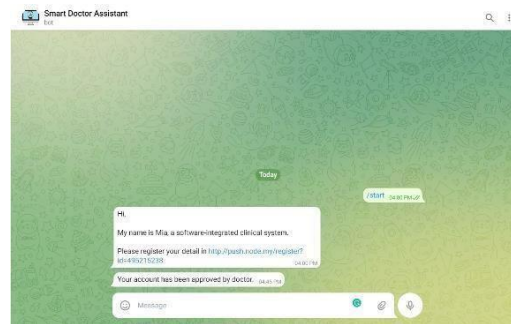


Figure 20: Approval from Doctor

Figure 20, demonstrates the approval process for the health officer. The left image displays the personal information provided by the patient for the hospital record. The medical officer can review the patient's profile and approve their registration, as shown in the right image. The notification sent to the patient's Telegram confirms that their registration has been accepted by the medical officer. Once approved, the patient can begin taking readings for body temperature, heart rate, and oxygen levels.

3.7.5 Pairing Device



Figure 21: Pair patient with registered device

Device Management System

NO	DEVICE SERIAL NO.	PAIRING WITH	UPDATED AT	
1	MLX90614/MAX30102	syukri yahya	01/01/2023 02:18	Unpair

Figure 22: Successful pair the device

The diagram illustrates a website where medical officers can pair a registered device with a registered patient. This pairing allows the patient to receive sensor readings from both sensors exclusively through their Telegram BOT, preventing confusion with readings from other patients or medical officers. Figure 21 shows the interface for pairing the device with a registered patient, where medical officers can select patients who wish to take readings of their body parameters. Figure 22 displays the name of the patient who has been successfully matched with the device.

3.7.6 Self-Monitoring MAX30102 & MLX90614



Figure 23: MAX30102 Heart Rate and Pulse Oximetry



Figure 24: MLX 90614 Temperature Sensor

Based on figures 23 and 24, the patient is taking heart rate and oxygen saturation readings. Figure 23 shows the patient taking heart rate and oxygen saturation readings. When placing the index finger on the sensor, the position is important. It functions as a sensor, reading the reading of our blood condition. Meanwhile, in figure 24, the patient is taking a temperature reading with the MLX90614 temperature sensor. The patient directs the temperature sensor to the forehead, getting as close to the sensor as possible.

3.7.7 Data Update on Telegram and Website

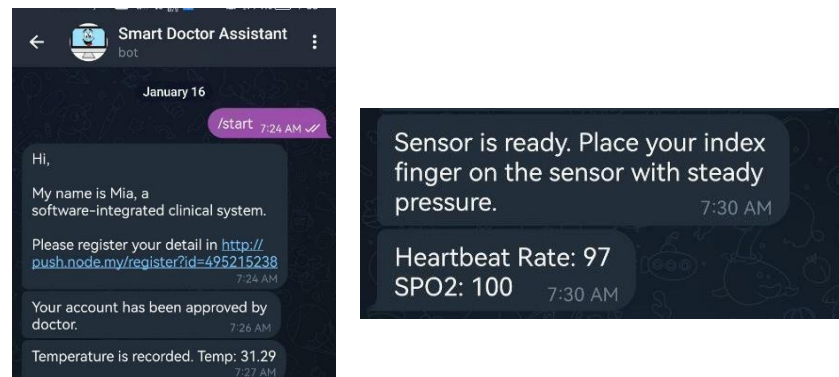


Figure 25: Sensor Reading on Telegram BOT

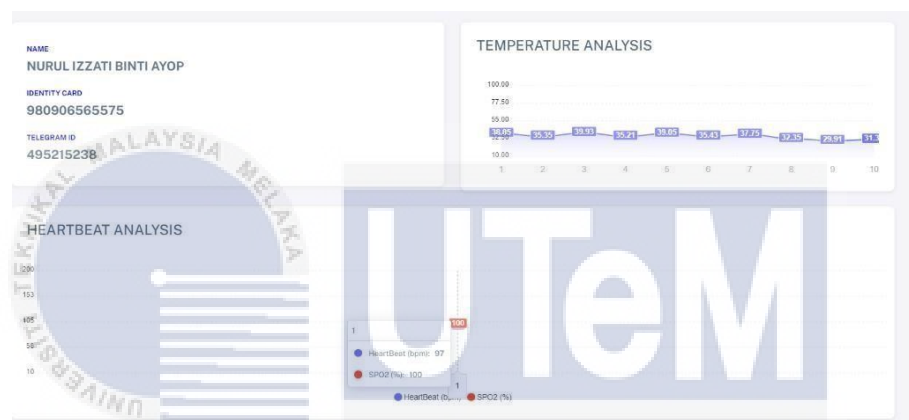


Figure 26: Sensor Reading on Website

After the medical officer pairs the device with the patient, the patient can begin self-monitoring by taking readings of their heart rate, oxygen level, and body temperature. These readings will be displayed exclusively on the patient's Telegram BOT, as shown in figure 25. In addition to being displayed on the patient's TelegramBOT, these readings will also be recorded and displayed on the website that is operated by the medical officer as shown in figure 26. This allows the medical officer to track the patient's readings and monitor their health remotely.

3.7.8 Doctor Analysis the History Patient with Sensor Reading and Give a Health Status

Patient Information

Name
NURUL IZZATI AYOP

Telegram ID
495215238

Age
25

Identity Card No.
980906565574

Mobile No.
01110009806

Address
DT526, Jalan Bukit Tambun Perdana 10, Taman Bukit Tambun, 76100, Durian Tunggal, Melaka

Health Information

Diabetes
☒ No
☐ Yes

High Blood Pressure
☒ No
☐ Yes

Asthma
☒ No
☐ Yes

Had Operation?
☒ No
☐ Yes

Fever?
☒ No
☐ Yes

Cough?
☒ No
☐ Dry Cough
☐ Cough With Mucus
☐ Normal Cough

Shortness Of Breath?
☒ No
☐ Yes

Fatigue
☒ No
☐ Yes

Body Aches
☒ No
☐ Yes

Loss of Smell / Taste
☒ No
☐ Yes

Sore Throats
☒ No
☐ Yes

Diarrhea
☒ No
☐ Yes

Nausea or Vomiting
☒ No
☐ Yes

How many days you have been sick?
☒ Less than 7 days
☐ More than 7 days

Heart Diseases
☒ No
☐ Yes

Other diseases
No.

[Monitor](#) [Message](#) [Edit Patient](#)

Figure 27: Patient Information Interface

After the patient completes self-monitoring, the medical officer will analyze their health based on the information provided by the patient as shown in figure 27. There are several ways to test for COVID-19, including RT-PCR, antigen test, serological test, and evaluating the symptoms experienced by the patient. Patients who may be considered positive for COVID-19 include those who show symptoms such as fever, dry cough, and shortness of breath. If a patient has underlying health conditions, they may be at higher risk of being infected with the COVID-19 virus. Based on this analysis, medical officers will notify patients through the Telegram BOT platform using their Telegram ID whether they are positive or negative for COVID-19. If a patient is found to be positive, they will be advised to self-quarantine and informed by the medical officer on the next steps to take. If a patient is negative for COVID-19, the medical officer will also notify them through the Telegram BOT platform. The analysis will be communicated through a link, which the patient must click on to access the result of their reading.

3.7.9 Patient Receive Result on the Telegram



Figure 28: Patient receive result from medical officer.

Based on figure 28 shows a display from the patient's Telegram. The display shows the results of the medical officer's research, which says that this patient is not infected with the COVID-19 virus. In the results section, there are two answer options, namely whether the patient is positive for the COVID-19 virus or negative for the COVID-19 virus. If the patient is found to be positive for COVID-19, the medical officer will send notification messages saying that the health results are positive for the COVID-19 virus, and the patient is asked to self-quarantine for 10 days to avoid contracting the COVID-19 virus with others. The medical officer will monitor the symptoms faced by the patient and discuss the treatment that needs to be carried out by the positive patient for COVID-19. The second answer is negative for COVID-19, where the patient does not need to undergo quarantine for 10 days, rather the patient needs to take precautions to prevent the spread of the COVID-19 virus to others. This message will be sent through the patient's Telegram BOT.

CHAPTER 4



In this section or chapter, the results and discussion will present the results obtained from the project. There will be an analysis and comparison of the result obtained using the device developed in the project and devices available in the market. The comparison will be presented through the use of graphs and relevant tables. This comparison will provide insight into the performance of the device developed in the project compared to similar devices available in the market.

4.1 General Measurement



Figure 29: General Measurement

Figure 29 shows the display of readings taken when there is no paired device. The activity sensor interface displays reading for heart rate, oxygen saturation, and body temperature. On the left is the wireless monitoring interface, which shows that the device is connected to the IP address of a mobile phone. The bottom part of the interface displays a graph showing readings for heart rate and oxygen saturation. The blue line represents the reading taken by the MAX 30102 sensor, which is the heart rate sensor, while the red line represents the reading for oxygen level.

4.2 Result for MAX 30102 (Heart Rate and Pulse Rate Sensor)

Table 5: Data of heart beat rate by using IoT device and Market Device

NO	Name	Hb rate sensor (bpm)	Hb rate market (bpm)	Error \pm	% Of Error
1	Nurul Izzati	97	84	0.1547	15.47
2	Dalinie Emysha	91	98	0.0714	7.14
3	Nur Najihah	99	110	0.1	10
4	Arina Amerah	82	90	0.0888	8.88
5	Nurul Safiqah	92	95	0.0316	3.158
6	Najlaa	81	71	0.1408	14.08
7	Nur Ismahani	97	89	0.0898	8.98
8	Syahira	94	91	0.0329	3.29
9	Awatif Maliq	88	71	0.2394	23.94
10	Elia Hani	93	87	0.0689	6.89

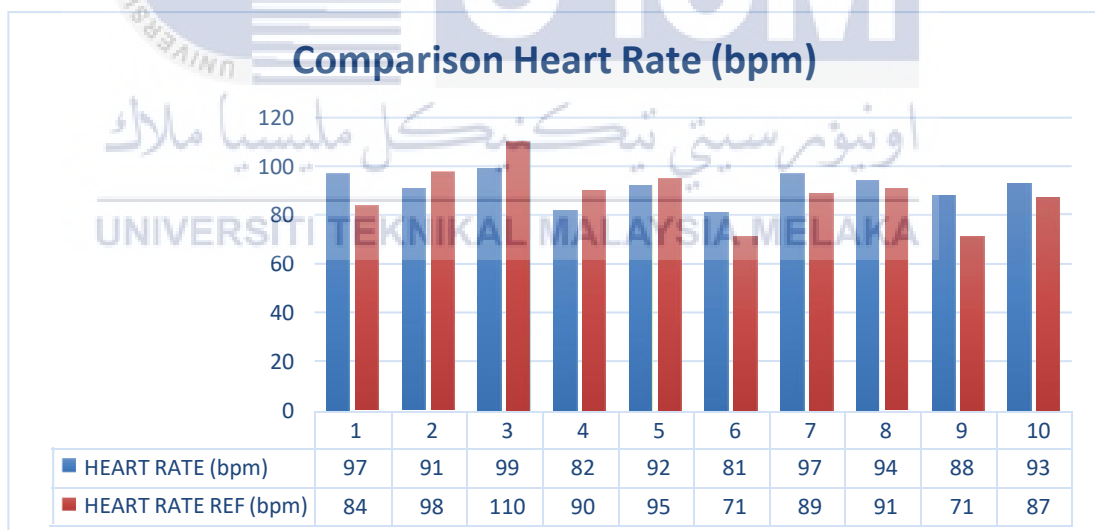


Figure 30: Comparison Column Chart between IoT Device and Market Device

Table 5 depicts a total of ten people as a project analysis. This information is derived from reading IoT Based Device for Self-Monitoring of COVID-19. Patients and devices are available on the market. This comparison is performed to ensure that the readings obtained with this device are accurate.

The first patient's heart beat rate data from this project's device was 97 bpm, while it was 80 bpm from the market device. As much as ± 0.1547 for precision. While the second patient's heart rate reached 91 bpm compared with the market device's 98 bpm with an accuracy of ± 0.0714 . The third patient's heart rate reached 99 bpm, and the market device is 110 with an accuracy of ± 0.1 . The fourth patient's heart rate reached 82 bpm and the market device is 90 bpm with accuracy ± 0.0888 . The fifth patient reached a heart rate of 92 bpm and the market device is 95 bpm with an accuracy of ± 0.0316 . The sixth patient's heart rate reached 81 bpm, and the market device was around 71 with an accuracy of 0.1408. The seventh patient's, heart rate reached around 97 and the market device is 89 with an accuracy of ± 0.0898 . The eight patient's heart rates reached 94 bpm and the market device is around 91 bpm with an accuracy of ± 0.0329 . The ninth patient's heart rate reached 88 and the market device is around 71 bpm with an accuracy of ± 0.2394 . The tenth patient's heart rate reached around 93 and the market device is 87 with an accuracy of ± 0.0689 .

Figure 30 shows the comparison for every patient by using the data from the project's device and market device. This comparison uses the column chart to see the comparison between the two data. The blue graph is for the heart rate (bpm) from the project's device and the red graph is for the heart rate from the market device.

4.3 Result for Oxygen Saturation by using MAX 30102

Table 6: Data of oxygen saturation by using IoT device and Market Device

NO	Name	Pulse Oximeter Sensor (SpO2)	Pulse Oximeter Reference (SpO2)	Error \pm	% Of Error
1	Nurul Izzati	100	97	0.0309	3.092
2	Dalinie Emysha	100	96	0.0416	4.167
3	Nur Najihah	100	110	0.0909	9.0909
4	Arina Amerah	97	98	0.0102	1.0204
5	Nurul Safiqah	100	95	0.0526	5.263
6	Najlaa	81	96	0.15625	15.625
7	Nur Ismahani	82	96	0.1458	14.58
8	Syahira	100	98	0.0204	2.0408
9	Awatif Maliq	100	97	0.0309	3.0927
10	Elia Hani	100	99	0.0101	1.0101

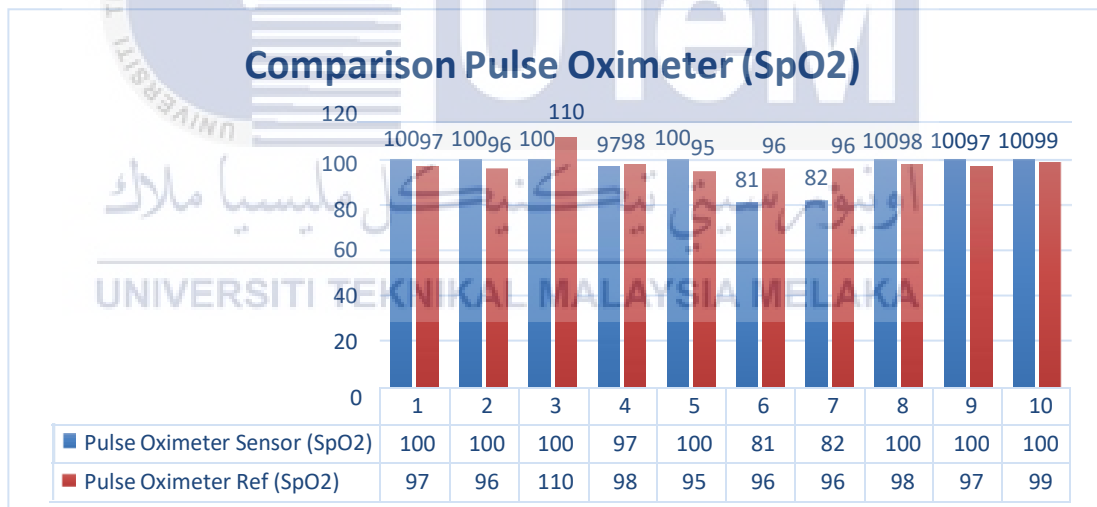


Figure 31: Comparison Column Chart between IoT Device and Market Device

Table 6 shows the data of oxygen saturation by using IoT Based Device for Self-monitoring of COVID-19 patients and a market device which is a pulse oximeter. There are 10 people as analysis persons for both devices.

The first patient's oxygen saturation data from this project's device was 100 SpO₂, while it was 97 SpO₂ from the market device. As much as ± 0.0309 for precision. While the second patient's oxygen saturation reached 100 SpO₂ compare with the market device's 96 SpO₂ with an accuracy of 0.0416. The third patient's oxygen saturation reached 100 SpO₂, and the market device is 110 SpO₂ with an accuracy of ± 0.0909 . The fourth patient's oxygen saturation reached 97 SpO₂ and the market device is 98 SpO₂ with an accuracy of ± 0.0102 . The fifth patient reached the oxygen saturation of 100 SpO₂ and the market device is 95 SpO₂ with an accuracy of 0.0526. The sixth patient's oxygen saturation reached 81 SpO₂, and the market device had around 96 SpO₂ with an accuracy of 0.15625. The seventh patient's, oxygen saturation reached around 82 SpO₂ and the market device is 96 SpO₂ with an accuracy of ± 0.1458 . The eight patient's oxygen saturation reached 100 SpO₂ and the market device is around 98 SpO₂ with an accuracy of ± 0.0204 . The ninth patient's oxygen saturation reached 100 SpO₂ and the market device is around 97 SpO₂ with an accuracy of ± 0.0309 . The tenth patient's oxygen saturation reached around 100 SpO₂ and the market device is 99 SpO₂ with an accuracy of ± 0.0101 .

Figure 31 shows the comparison for every patient by using the data from the project's device and market device. This comparison uses the column chart to see the comparison between the two data. The blue graph is for the pulse oximeter sensor (SpO₂) from the project's device and the red graph is for the pulse oximeter sensor reference (SpO₂) from the market device.

4.4 Result for temperature by using MLX 90614 Temperature Sensor

Table 7: Data of Temperature Sensor by using MLX 90614

NO	Name	Temperature Sensor (°C)	Thermometer (°C)	Error ±	% Of Error
1	Nurul Izzati	37.75	37.5	0.0066	0.66
2	Dalinie Emysha	36.55	37.9	0.0356	3.5620
3	Nur Najihah	37.27	36.1	0.03241	3.2409
4	Arina Amerah	36.39	36.5	0.003013	0.3014
5	Nurul Safiqah	37.09	37.3	0.00563	0.5630
6	Najlaa	37.01	37.0	0.0002702	0.0270
7	Nur Ismahani	36.61	36.8	0.005163	0.5163
8	Syahira	36.7	36.6	0.0027322	0.2732
9	Awatif Maliq	37.19	36.5	0.0189	1.8904
10	Elia Hani	36.7	36.3	0.01101	1.10193

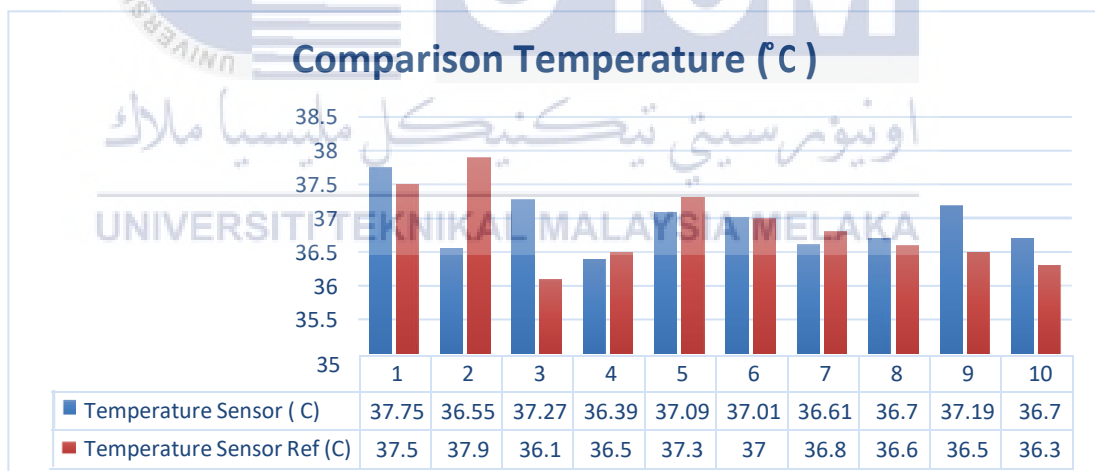


Figure 32: Comparison Column Chart between IoT Device and Market Device

Table 7 shows the data of the temperature sensor by using MLX 90614 and the thermometer that is usually used to take temperature during fever. Table 7 represents 10 people as an analysis person for this project.

The temperature for the first patient is 37.75 (°C) using the project's device, while 37.5 (°C) using the thermometer. The accuracy between these two devices is ± 0.0066 . The second patient is 36.55 (°C) and 37.9 (°C) for the market device with an accuracy of ± 0.0356 . While the temperature for the third patient is 37.27 (°C) and the market is 36.1 (°C) with accuracy ± 0.03241 . The temperature for the fourth patient is 36.39 (°C) and the thermometer is 36.5 (°C) with an accuracy of ± 0.003013 . The temperature fifth patient's 37.09 (°C) and the thermometer is 37.3 (°C) with an accuracy of ± 0.00563 . The temperature for the sixth patient is 37.01 (°C) and the thermometer was 37.0 (°C) with an accuracy of ± 0.0002702 . The temperature for the seventh patient is 36.61 (°C) and the thermometer is 36.8 (°C) with an accuracy of ± 0.005163 . The temperature for eight patients is 36.7 (°C) and the thermometer is 36.6 (°C) with accuracy ± 0.0027322 . The temperature sensor for the ninth patient is 37.19 (°C) and the thermometer 36.5 (°C) with accuracy ± 0.0189 and the tenth patient's is 36.7 (°C) and the thermometer 36.3 (°C) with accuracy ± 0.01101 .

Figure 32 shows the column chart for the two devices of the temperature sensor. The blue column chart is for the temperature sensor using MLX 90614 while the red column chart is for the thermometer which is the market device.

4.5 Project Significance

The importance of the project for this project is divided into three, namely cost savings, user-friendliness, and time savings. This project makes two sensors in one device. Whereas the market only has one function for one device. An IoT-Based Device for Self-Monitoring of COVID-19 patients has two sensors in one device. This can save the patient expenses from buying both sensors.

This project also uses the Telegram BOT platform which makes it easy for users to use it by simply installing the Telegram application. The Telegram application is available for both Android and IOS operating systems. Telegram BOT is used as a medium of communication between the patient and the medical team to diagnose the patient.

This project can also save patients time by not having to wait in long queues while at the clinic or hospital. As we know, if many patients come at one time, it requires patients to queue long before seeing the doctor for further examination. This causes the patient to waste time seeing the doctor for just a few minutes. So, this IoT-Based Device, can save patients' time and patients can communicate online without having to wait in long queues.

4.6 Project Sustainable Development Goals

The Sustainable Development Goals project has 17 global goals that have been set. The SDGs cover social and economic development issues including poverty, hunger, health, education, global warming, gender equality, water, sanitation, energy, urbanization, the environment, and social justice.

The Sustainable Development Goal related to IoT Based Devices for Self-Monitoring of COVID-19 patients project is sustainable development goal 3, which is Good Health and Well Being. One of the objectives of sustainable development goal 3 is to reduce the number of new cases and deaths from malaria and other major diseases.

Self-monitoring for COVID-19 can involve individuals taking steps to monitor their health for signs of COVID-19 infection such as checking body temperature and monitoring symptoms such as cough, fever, and difficulty breathing. By performing self-monitoring individuals can protect their health and the health of those around them, helping to achieve SDG 3 goals.



Figure 33: Sustainable Development Goals

4.7 Comparison of IoT-Based Device for Self-monitoring of COVID-19 Patient

Each project has its advantages over the previous one. IoT Based Devices for Self-monitoring of COVID-19 Patients have advantages over the previous project.

The IoT Based Device for Self-monitoring of COVID-19 Patients has three advantages where the first advantage of this project is having its Telegram BOT where this Telegram application is available for both operating systems such as Android and IOS. When compared to the MYSEJAHTERA platform, the MYSEJAHTERA application requires program code for both operating systems, namely Android and IOS.

The second advantage is that sensor readings are automatically displayed on Telegram BOT and continue to update on the website, where on this website the medical officer will analyze or diagnose the patient based on the patient's health record. Compared to the MYSEJAHTERA application, patients are required to manually update sensor readings into the MYSEJAHTERA platform.

The third advantage is there are two sensors in one device, which this sensor works a reading heart rate, oxygen saturation, and body temperature. This can help people to buy a device that has 2 different sensors which can make them save costs from buying two different sensors with one sensor and one function. By using this device, the cost of a project can be cut into half discount from the original price.

CHAPTER 5



5.1 Conclusion

In conclusion, this project can assist patients in monitoring themselves at home, also known as self-monitoring. This project includes two sensors, the first of which is the MAX 30102, a sensor that measures heart rate and oxygen saturation. The second sensor is the MLX90614, which measures the temperature of the human body. These two sensors take general readings that are always requested by the hospital or clinic. When patients are sick or former COVID-19 patients, these two sensors can help them understand their condition.

This project also created a Telegram BOT to help the patient communicate with medical officers over the phone. Telegram BOT was chosen as a two-way communication medium because it does not require separate programmed code for the Android and IOS operating systems. The Telegram application is available in the play store and Appstore for both of these operating systems.

Finally, these projects aim to assist or reduce the increasing number of deaths. The existence of this project will help to reduce the number of COVID-19 patient deaths as well as the increasing prevalence of COVID-19 infections. As a result of this project, medical officers and patients can be breaking the COVID-19 chain.

5.2 Future Work

The project has four future works that can be renewed or upgraded from this project. The future work is to make it easier for the community to use the IoT Based Device for Self-Monitoring of COVID-19 patients.

The first future work is for the patient whereby the patient can scan QR Code provided with the device and the QR code will automatically direct them to their Telegram account. This can be a security for the user of the Telegram application especially the user of Telegram BOT “Smart Monitor Doctor”. All the information only be shared through their Telegram application.

The second future project is to have the device automatically connect to the strong WiFi signal that is available at the time. The third future project is to obtain accurate readings for both sensors so that this device’s readings match those of other devices. The last future work is this type of project can be applied to other applications related to health or self-monitoring at home. Also in the future, the final project is can be designed simply and easier for the patient to use.

REFERENCES

- [1] M. M. Khan, S. Mehnaz, A. Shaha, M. Nayem, and S. Bourouis, "IoT-Based Smart Health Monitoring System for COVID-19 Patients," *Comput. Math. Methods Med.*, vol. 2021, 2021, doi: 10.1155/2021/8591036.
- [2] O. Taiwo and A. E. Ezugwu, "Smart healthcare support for remote patient monitoring during covid-19 quarantine," *Informatics Med. Unlocked*, vol. 20, p. 100428, 2020, doi: 10.1016/j.imu.2020.100428.
- [3] M. M. Islam, A. Rahaman, and M. R. Islam, "Development of Smart Healthcare Monitoring System in IoT Environment," *SN Comput. Sci.*, vol. 1, no. 3, pp. 1–11, 2020, doi: 10.1007/s42979-020-00195-y.
- [4] G. Casalino, G. Castellano, and G. Zaza, "A mHealth solution for contact-less self-monitoring of blood oxygen saturation," *Proc. - IEEE Symp. Comput. Commun.*, vol. 2020-July, 2020, doi: 10.1109/ISCC50000.2020.9219718.
- [5] S. S. Vedaiei *et al.*, "COVID-SAFE: An IoT-based system for automated health monitoring and surveillance in post-pandemic life," *IEEE Access*, vol. 8, pp. 188538–188551, 2020, doi: 10.1109/ACCESS.2020.3030194.

- [6] A. I. Siam *et al.*, “Secure Health Monitoring Communication Systems Based on IoT and Cloud Computing for Medical Emergency Applications,” *Comput. Intell. Neurosci.*, vol. 2021, 2021, doi: 10.1155/2021/8016525.
- [7] V. Yeri and D. C. Shubhangi, “IoT based Real Time Health Monitoring,” *Proc. 2nd Int. Conf. Inven. Res. Comput. Appl. ICIRCA 2020*, pp. 980–984, 2020, doi: 10.1109/ICIRCA48905.2020.9183194.
- [8] G. Basaranoglu, M. Bakan, T. Umutoglu, S. U. Zengin, K. Idin, and Z. Salihoglu, “Comparison of SpO2 values from different fingers of the hands,” *Springerplus*, vol. 4, no. 1, pp. 2–4, 2015, doi: 10.1186/s40064-015-1360-5.
- [9] N. Arya, “IoT based Smart Health Monitoring Band for COVID-19 Patients,” vol. 4, no. 7, pp. 50–53, 2021.
- [10] A. S. Bennet Praba, Anuj Kakar, “IoT Based Handheld Smart Health Monitoring System for,” *Syst. Rev. Pharm.*, vol. 12, no. 5, pp. 239–242, 2021.
- [11] M. Otoom, N. Otoum, M. A. Alzubaidi, Y. Etoom, and R. Banihani, “An IoT-based framework for early identification and monitoring of COVID-19 cases,” *Biomed. Signal Process. Control*, vol. 62, no. April, p. 102149, 2020, doi: 10.1016/j.bspc.2020.102149.
- [12] M. Mapelli, C. Vignati, P. Gugliandolo, D. Fumagalli, and P. Agostoni, “Feasibility of remote home monitoring with a T-shirt wearable device in post-recovery COVID-19 patients,” *J. Cardiovasc. Med. (Hagerstown)*, vol. 22, no. 11, pp. 860–863, 2021, doi: 10.2459/JCM.0000000000001165.
- [13] R. Stojanovic, A. Skraba, and B. Lutovac, “A Headset Like Wearable Device

- to Track COVID-19 Symptoms,” *2020 9th Mediterr. Conf. Embed. Comput. MECO 2020*, pp. 8–11, 2020, doi: 10.1109/MECO49872.2020.9134211.
- [14] K. Roy and S. Sarkar, “Pr ep rin t n ot pe er r ev we d Pr ep rin t n ot pe er r,” 2020.
- [15] S. N. S. Al-Humairi and A. A. A. Kamal, “Design a smart infrastructure monitoring system: a response in the age of COVID-19 pandemic,” *Innov. Infrastruct. Solut.*, vol. 6, no. 3, pp. 1–10, 2021, doi: 10.1007/s41062-021-00515-y.
- [16] P. W. Rusimamto, R. Harimurti, Endryansyah, Y. Anistyasari, and L. Anifah, “Design and Implementation of Thermal Body System Employing Thermal Sensor MLX 90614 for Covid-19 Symptoms Early Detector,” vol. 196, no. Ijce, pp. 317–321, 2020, doi: 10.2991/aer.k.201124.058.
- [17] F. Jamil, S. Ahmad, N. Iqbal, and D. H. Kim, “Towards a remote monitoring of patient vital signs based on iot-based blockchain integrity management platforms in smart hospitals,” *Sensors (Switzerland)*, vol. 20, no. 8, 2020, doi: 10.3390/s20082195.
- [18] I. Journal, “IRJET-Human Health Monitoring System”.
- [19] A. Kaur and A. Jasuja, “Health monitoring based on IoT using Raspberry PI,” *Proceeding - IEEE Int. Conf. Comput. Commun. Autom. ICCCA 2017*, vol. 2017-Janua, pp. 1335–1340, 2017, doi: 10.1109/CCAA.2017.8230004.
- [20] S. Mahapatra, V. Kannan, S. Seshadri, and V. Ravi, “An IoT-Based Wristband for Automatic People Tracking , Contact Tracing and Geofencing for COVID-

19,” 2022.



APPENDICES

Source Code for MAX 30102

(Heart Rate and Pulse Oximetry Sensor)

```
#include <WiFi.h>
#include <Wire.h>
#include "MAX30105.h"
#include "spo2_algorithm.h"
```

```
#include "heartRate.h"
```

```
// WiFi Credentials
```

```
const char* ssid = "Norsiah Family";
```

```
const char* password = "norsiah67";
```

```
// Server Node Telegram
```

```
const char* host = "push.node.my";
```

```
MAX30105 particleSensor;
```

```
const byte RATE_SIZE = 4; //Increase this for more averaging. 4 is good.
```

```
byte rates[RATE_SIZE]; //Array of heart rates
```

```
byte rateSpot = 0;
```

```
long lastBeat = 0; //Time at which the last beat occurred
```

```
float beatsPerMinute;
```

```
int beatAvg;
```

```
int32_t bufferLength;
```

```
int32_t spo2;
```

```
int8_t validSPO2;
```

```
int32_t heartRate;
```

```
int8_t validHeartRate;
```

```
uint32_t irBuffer[100];
```

```
uint32_t redBuffer[100];
```

```
int isSent = 0;
```

```
WiFiClient client;
```

```
void setup()
```

```
{
```

```
  Serial.begin(115200);
```

```

Serial.println("Initializing...");

// Connect wifi first, and inform to telegram
wifi_connect();

// check sensor and send sensor status
heartbeat_ready();

particleSensor.setup(); //Configure sensor with default settings

particleSensor.setPulseAmplitudeRed(0x0A); //Turn Red LED to low to indicate
sensor is running
particleSensor.setPulseAmplitudeGreen(0); //Turn off Green LED
}

void loop()
{
    bufferLength = 100;

    for (byte i = 0; i < bufferLength; i++) {
        while (particleSensor.available() == false)
            particleSensor.check();

        redBuffer[i] = particleSensor.getRed();
        irBuffer[i] = particleSensor.getIR();
        particleSensor.nextSample();
    }
}

```



```

Serial.print(F("count="));

Serial.print(i, DEC);

Serial.print(F(", red="));

Serial.print(redBuffer[i], DEC);

Serial.print(F(", ir="));

Serial.println(irBuffer[i], DEC);

}

```

```

    maxim_heart_rate_and_oxygen_saturation(irBuffer,  bufferLength,  redBuffer,
    &spo2, &validSPO2, &heartRate, &validHeartRate);

```

```

while (1)
{
    //dumping the first 25 sets of samples in the memory and shift the last 75 sets of
    samples to the top
    for (byte i = 25; i < 100; i++)
    {
        redBuffer[i - 25] = redBuffer[i];

        irBuffer[i - 25] = irBuffer[i];
    }

    //take 25 sets of samples before calculating the heart rate.

    for (byte i = 75; i < 100; i++)
    {
        while (particleSensor.available() == false) //do we have new data?

            particleSensor.check(); //Check the sensor for new data
    }
}

```

```
//    digitalWrite(readLED, !digitalRead(readLED));

redBuffer[i] = particleSensor.getRed();

irBuffer[i] = particleSensor.getIR();

particleSensor.nextSample(); //We're finished with this sample so move to next
sample
```

```
//send samples and calculation result to terminal program through UART
```

```
//    Serial.print(F("red="));

//    Serial.print(redBuffer[i], DEC);

//    Serial.print(F(", ir="));

//    Serial.print(irBuffer[i], DEC);

//

//    Serial.print(F(", HR="));

//    Serial.print(heartRate, DEC);

//

//    Serial.print(F(", HRvalid="));

//    Serial.print(validHeartRate, DEC);

//

//    Serial.print(F(", SPO2="));

//    Serial.print(spo2, DEC);

//

//    Serial.print(F(", SPO2Valid="));

//    Serial.println(validSPO2, DEC);
```

```

if (i == 99 && validHeartRate == 1 && validSPO2 == 1 && isSent == 0) {

    Serial.print(F("red="));

    Serial.print(redBuffer[i], DEC);

    Serial.print(F(", ir="));

    Serial.print(irBuffer[i], DEC);


    Serial.print(F(", HR="));

    Serial.print(heartRate, DEC);


    Serial.print(F(", HRvalid="));

    Serial.print(validHeartRate, DEC);

    Serial.print(F(", SPO2="));

    Serial.print(spo2, DEC);

    Serial.print(F(", SPO2Valid="));

    Serial.println(validSPO2, DEC);


    monitor_heartbeat_spo();

    isSent = 1;

}


if (validHeartRate == 0) {

    isSent = 0;

}

```

```
}
```

```
    maxim_heart_rate_and_oxygen_saturation(irBuffer, bufferLength, redBuffer,
    &spo2, &validSPO2, &heartRate, &validHeartRate);
```

```
}
```

```
// // Infrared sensor value
```

```
// long irValue = particleSensor.getIR();
```

```
//
```

```
// //Serial.println(irValue);
```

```
//
```

```
// // if beats detected!
```

```
// if (checkForBeat(irValue) == true)
```

```
// {
```

```
//    long delta = millis() - lastBeat;
```

```
//    lastBeat = millis();
```

```
//
```

```
// // convert beats in minute
```

```
//    beatsPerMinute = 60 / (delta / 1000.0);
```

```
//
```

```
// // calculate average if bpm more than 20 and lower than 255
```

```
// if (beatsPerMinute < 255 && beatsPerMinute > 20)
```

```
// {
```

```
//    rates[ratesSpot++] = (byte)beatsPerMinute;
```

```
//    ratesSpot %= RATE_SIZE;
```

```
//
```

```
//    beatAvg = 0;
```

```

//    for (byte x = 0 ; x < RATE_SIZE ; x++)

//        beatAvg += rates[x];

//    beatAvg /= RATE_SIZE;

// }

// }

//

// // show bpm if bpm and average more than 0

// if (beatsPerMinute > 0 && beatAvg > 0) {

//

//    Serial.print("IR=");

//    Serial.print(irValue);

//    Serial.print(", BPM=");

//    Serial.print(beatsPerMinute);

//    Serial.print(", Avg BPM=");

//    Serial.print(beatAvg);

//

// // show error when finger not detected

// if (irValue < 50000) {

//    Serial.print(" No finger?");

// }

//

//    Serial.println();

// // delay(1000);

// }

}

```

```

void monitor_heartbeat_spo() {

    if (client.connect(host, 80)) {

        String url = "/iot/esp32";

        url += "?code=CHECK";

        url += "&hr=";

        url += heartRate;

        url += "&spo=";

        url += spo2;

        Serial.println(url);

        client.print(String("GET ") + url + " HTTP/1.1\r\nHost: " + host +
"\r\nConnection: close\r\n\r\n");

        unsigned long timeout = millis();
        while (client.available() == 0) {
            if (millis() - timeout > 5000) {

                Serial.println(">>> Client Timeout !");

                client.stop();

                return;

            }
        }

        while (client.available()) {

            String line = client.readStringUntil('\r');

            Serial.print(line);

```

```

    }

}

void wifi_connect() {

    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {

        delay(500);
        Serial.print(".");
    }

    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());

    // Use WiFiClient class to create TCP connections

    const int httpPort = 80;

    if (!client.connect(host, httpPort)) {

        Serial.println("connection failed");

        return;

    }

```

```
// We now create a URI for the request

String url = "/iot/esp32";

url += "?code=";

url += "WIFI_STATUS";

url += "&ssid=";

url += urlencode(ssid);

url += "&ip=";

url += WiFi.localIP().toString();

Serial.println(url);

// This will send the request to the server
client.print(String("GET ") + url + " HTTP/1.1\r\n" +
    "Host: " + host + "\r\n" +
    "Connection: close\r\n\r\n");

unsigned long timeout = millis();
while (client.available() == 0) {
    if (millis() - timeout > 5000) {
        Serial.println(">>> Client Timeout !");
        client.stop();
        return;
    }
}

// Read all the lines of the reply from server and print them to Serial
```



```

while (client.available()) {

    String line = client.readStringUntil('\r');

    Serial.print(line);

}

}

```

```

String urlencode(String str) {

    String encodedString = "";

    char c;

    char code0;
    char code1;
    char code2;

    for (int i = 0; i < str.length(); i++) {
        c = str.charAt(i);
        if (c == '\\') {
            encodedString += '%';
            encodedString += '2';
            encodedString += '0';
        } else if (isalnum(c)) {
            encodedString += c;
        } else {
            code1 = (c & 0xf) + '0';
            if ((c & 0xf) > 9) {
                code1 = (c & 0xf) - 10 + 'A';
            }

```

```

c = (c >> 4) & 0xf;

code0 = c + '0';

if (c > 9) {

    code0 = c - 10 + 'A';

}

code2 = '\0';

encodedString += '%';

encodedString += code0;

encodedString += code1;

//encodedString+=code2;

}

yield();

}

return encodedString;

}

```

```

void heartbeat_ready() {

```

```

    // Initialize sensor

```

```

    if (!particleSensor.begin(Wire, I2C_SPEED_FAST)) //Use default I2C port,
400kHz speed

```

```

{

```

```

    Serial.println("MAX30102 was not found. Please check wiring/power. ");

```

```

    while (1);

```

```

}

```

```

    Serial.println("Place your index finger on the sensor with steady pressure.");

```

```

if (client.connect(host, 80)) {

    String url = "/iot/esp32";

    url += "?code=HEARTBEAT";

    url += "&data=READY";

    client.print(String("GET ") + url + " HTTP/1.1\r\nHost: " + host +
"\r\nConnection: close\r\n\r\n");

    unsigned long timeout = millis();

    while (client.available() == 0) {

        if (millis() - timeout > 5000) {

            Serial.println(">>> Client Timeout !");

            client.stop();

            return;

        }

    }


}

}

}

}

```



Source Code for MLX90614 (Temperature Sensor)

```
#include <WiFi.h>

#include <Wire.h>

#include <Adafruit_MLX90614.h>


// WiFi Credentials

const char* ssid    = "Norsiah Family";

const char* password = "norsiah67";


// Server Node Telegram

const char* host = "push.node.my";

WiFiClient client;

Adafruit_MLX90614 mlx = Adafruit_MLX90614();


// previous temperature

double previous_temp = 0;


// difference temperature

double diff_temp;


void setup()

{

    Serial.begin(115200);

    Serial.println("Initializing...");
```

```
// Connect wifi first, and inform to telegram
```

```
wifi_connect();
```

```
while (!Serial);
```

```
if (!mlx.begin()) {
```

```
    Serial.println("Error connecting to MLX sensor. Check wiring.");
```

```
    while (1);
```

```
};
```

```
}
```

```
void loop()
```

```
{
```

```
    // Serial.print("Ambient temperature = ");
```

```
    // Serial.print(mlx.readAmbientTempC());
```

```
    // Serial.print("°C");
```

```
    // Serial.print(" ");
```

```
    // Serial.print("Object temperature = ");
```

```
    // Serial.print(mlx.readObjectTempC());
```

```
    // Serial.println("°C");
```

```
    //
```

```
    // Serial.print("Ambient temperature = ");
```



اونيورسيتي تيكنيكل مليسيا ملاكا

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

```
// Serial.print(mlx.readAmbientTempF());

// Serial.print("°F");

// Serial.print(" ");

// Serial.print("Object temperature = ");

// Serial.print(mlx.readObjectTempF());

// Serial.println("°F");

//

// Serial.println(".....");

delay(1000);
```

```
// if previous temperature was 0, then send data for initial data
if (previous_temp == 0) {
    previous_temp = mlx.readObjectTempC();
    Serial.println(previous_temp);
    send_temp(previous_temp);
}
UNIVERSITI TEKNIKAL MALAYSIA MELAKA
```

```
else {
```

```
    // if previous data is not equal to current temperature,
```

```
    // then save current temp into previous temp and
```

```
    // send data
```

```
    if (previous_temp != mlx.readObjectTempC()) {
```

```
        // if difference between previous and current more than 1.0
```

```

// then, send data

diff_temp = (previous_temp > mlx.readObjectTempC()) ? previous_temp -
mlx.readObjectTempC() : mlx.readObjectTempC() - previous_temp;

Serial.print("difference: ");

Serial.println(diff_temp);

if (diff_temp > 1) {

    // save current into previous temp

    previous_temp = mlx.readObjectTempC();

    Serial.println(previous_temp);

    // send data
    send_temp(previous_temp);

}

}

}

String urlencode(String str) {

    String encodedString = "";

    char c;

    char code0;

```

```

char code1;

char code2;

for (int i = 0; i < str.length(); i++) {

    c = str.charAt(i);

    if (c == ' ') {

        encodedString += '%';

        encodedString += '2';

        encodedString += '0';

    } else if (isalnum(c)) {

        encodedString += c;

    } else {

        code1 = (c & 0xf) + '0';

        if ((c & 0xf) > 9) {

            code1 = (c & 0xf) - 10 + 'A';

        }

        c = (c >> 4) & 0xf;

        code0 = c + '0';

        if (c > 9) {

            code0 = c - 10 + 'A';

        }

        code2 = '\0';

        encodedString += '%';

        encodedString += code0;

        encodedString += code1;

        //encodedString+=code2;

```



```

    }

    yield();

}

return encodedString;

}

```

```
void wifi_connect() {
```

```
    WiFi.begin(ssid, password);
```

```
    while (WiFi.status() != WL_CONNECTED) {
```

```
        delay(500);
```

```
        Serial.print(".");
```

```
    }
```

```
    Serial.println("");
```

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

```
    Serial.println("IP address: ");
```

```
    Serial.println(WiFi.localIP());
```

```
// Use WiFiClient class to create TCP connections
```

```
const int httpPort = 80;
```

```
if (!client.connect(host, httpPort)) {
```

```

Serial.println("connection failed");

return;

}

// We now create a URI for the request

String url = "/iot/esp32";

url += "?code=";

url += "WIFI_STATUS";

url += "&ssid=";

url += urlencode(ssid);

url += "&ip=";

url += WiFi.localIP().toString();

// This will send the request to the server
client.print(String("GET ") + url + " HTTP/1.1\r\n" +
              "Host: " + host + "\r\n" +
              "Connection: close\r\n\r\n");

unsigned long timeout = millis();

while (client.available() == 0) {

  if (millis() - timeout > 5000) {

    Serial.println(">>> Client Timeout !");

    client.stop();

    return;

  }

}

```

```
// Read all the lines of the reply from server and print them to Serial
```

```
while (client.available()) {
```

```
    String line = client.readStringUntil('\r');
```

```
    Serial.print(line);
```

```
}
```

```
}
```

```
void send_temp(double temp) {
```

```
    if (client.connect(host, 80)) {
```

```
        String url = "/iot/esp32";
```

```
        url += "?code=TEMP";
```

```
        url += "&temp=";
```

```
        url += temp;
```

```
        client.print(String("GET ") + url + " HTTP/1.1\r\nHost: " + host +  
"\r\nConnection: close\r\n\r\n");
```

```
        unsigned long timeout = millis();
```

```
        while (client.available() == 0) {
```

```
            if (millis() - timeout > 5000) {
```

```
                Serial.println(">>> Client Timeout !");
```

```
                client.stop();
```

```
                return;
```

```
            }
```

```
        }
```

```
    }
```



```
}
```

Source Code Arduino ESP 32 to Telegram in PHP

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use Illuminate\Http\Request;
```

```
use App\Http\Controllers\TelegramController as TG;
```

```
use App\Models\Sensors;
```

```
use App\Models\DeviceModel;
```

```
use App\Models\PatientModel;
```

```
class ESP32Controller extends Controller
```

```
{
```

```
    public function index(Request $request){
```

```
        if(!$request->has('code')){
```

```
            return 'error';
```

```
        }
```

```
    else {
```

```
        $code = $request->code;
```

```
        $data = $request->has('data') ? $request->data : '';
```

```
        $serial_number = 'MLX90614/MAX30102';
```

```

$device = new DeviceModel();

$device->get_serial_number($serial_number);

$patient = new PatientModel();

if($code == 'WIFI_STATUS'){

    $ip = $request->has('ip') ? $request->ip : "";

    $ssid = $request->has('ssid') ? $request->ssid : "";

```

```

    if($ssid == "" || $ip == ""){

```

```

        // dont send

```

```

    }

```

```

    else {

```

```

        Sensors::wifi()->save($ssid, $ip);

```

```

    if($device->hasPairing()){

```

```

        $patient = $patient->get_one($device->get_pairing_id());

```

```

        TG::reply($patient->telegram_id, "Device has been connected to
Internet with IP: " . $ip)->send();

```

```

    }

```

```

}

```

```

}

```

```

elseif($code == 'HEARTBEAT'){

```

```

    if($data == 'READY'){

```

```

        if($device->hasPairing()){

```

```

    $patient = $patient->get_one($device->get_pairing_id());

    TG::reply($patient->telegram_id, 'Sensor is ready. Place your index
finger on the sensor with steady pressure.')->send();

    }

}

}

elseif($code == 'TEMP'){

    if($request->has('temp')){

        $temp = $request->temp;

        if(!$device->hasPairing()){

            Sensors::temperature()->save($temp);

        }

        else {

            $patient = $patient->get_one($device->get_pairing_id());

            Sensors::Patient($patient->id)->temperature()->save($temp);

            TG::reply($patient->telegram_id, "Temperature is recorded. Temp:
" . $temp)->send();

        }

    }

}

}

elseif($code == 'CHECK'){

    if($request->has('hr', 'spo')){

```

```

$hr = $request->hr;

$spo = $request->spo;

if(!$device->hasPairing()){

    Sensors::heartbeat()-

    >save($hr, $spo);

}

else {

    $data = $patient->get_one($device->get_pairing_id());

    $age = $patient->get_age($data->ic_no);

    if($age == 0){

        TG::reply($data->telegram_id, 'Identity Card
        number was
        invalid. Please ask administrator to update your identity card number in
        dashboard!')-
        >send();

    }

    else {

        $max_hr = 220 - $age;

        $min_target_hr = 60 / 100 * $max_hr;

        $max_target_hr = 85 / 100 * $max_hr;

        $hr = $hr - $age;

        $hr_state = false;

```

```

if($hr > 50 && $hr < 90){

    $hr = rand(80, 95)
    $hr_state = true;

}

elseif($hr > 0 && $hr < 51){

    TG::reply($data->telegram_id, 'Error hardware
    reading!
    Please make sure to put your finger correctly.')->send();

}

elseif($hr > 110 && $hr < 120) {

    $hr = rand(96,100);
    $hr_state = true;

}

elseif($hr > 150){

    TG::reply($data->telegram_id, 'Error hardware
    reading!
    Please make sure to put your finger correctly.')->send();

}

if($hr_state){

    // TG::reply($data->telegram_id, 'HR: ' . $hr . '
    (Target: ' .
    $min_target_hr . ' - ' . $max_target_hr . ')')->send();

    Sensors::Patient($data->id)->heartbeat()-
    >save($hr, $spo);TG::reply($data-

```



```

        >telegram_id, "Heartbeat Rate: " . $hr .
"\nSPO2: " . $spo)->send();
    }

    }

    }

    }

    return 'received';

}

```

