DEVELOPMENT OF AUTOMATIC HEALTH MONITORING SYSTEM USING ARDUINO

SYAZWANI BINTI ROSMADI

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

DEVELOPMENT OF AUTOMATIC HEALTH MONITORING SYSTEM USING ARDUINO

SYAZWANI BINTI ROSMADI

THIS REPORT IS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Bachelor of Electronics Engineering Technology (Telecommunications) with Honours

FACULTY OF ELECTRONICS AND COMPUTER TECHNOLOGY AND ENGINEERING

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2025

UNIVERSITI TEKNIKAL MALAYSIA MELAKA
FAKULTI TEKNOLOGI DAN KEJURUTERAAN ELEKTRONIK DAN KOMPUTER

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : DEVELOPMENT OF AUTOMATIC HEALTH MONITORING SYSTEM USING ARDUINO

Sesi Pengajian : 2024/2025

Saya SYAZWANI BINTI ROSMADI mengaku membenarkan laporan Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Laporan adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. Sila tandakan (✓):

☐ **SULIT*** (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

☐ **TERHAD*** (Mengandungi maklumat terhad yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan.

☑ **TIDAK TERHAD**

Disahkan oleh:

_________________________ ________________________________

MOHD KHANAPIAH BIN NOR
Pensyarah
Fakulti Teknologi Dan Kejuruteraan Elektronik Dan Komputer (FTKEK)
Universiti Teknikal Malaysia Melaka (UTeM)

Alamat Tetap: ....

Tarikh : 17 Januari 2025 Tarikh : 24 / 1 / 2025

*CATATAN: Jika laporan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh laporan ini perlu dikelaskan sebagai SULIT atau TERHAD.

DECLARATION

I declare that this project report entitled "Development of Automatic Health Monitoring System Using Arduino" is the result of my own research except as cited in the references. The project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : ______________________

Student Name : SYAZWANI BINTI ROSMADI

Date : 17 JANUARI 2025

APPROVAL

I hereby declare that I have checked this project report and in my opinion, this project report is adequate in terms of scope and quality for the award of the degree of Bachelor of Electronics Engineering Technology (Telecommunications) with Honours.

Signature	:	
Supervisor Name	:	MOHD KHANAPIAH BIN NOR
Date	:	24 / 1 / 2025
Signature	:	
Co-Supervisor Name (if any)	:	
Date	:	

DEDICATION

I dedicate my bachelor's degree project with heartfelt gratitude to my mother, Zaini Binti Jaaffar and my family. Their unwavering support,valuable advice,and financial assistance have been instrumental in shaping me into an independent and determined individual. Their life lessons have inspired me to persevere through challenges ,reinforcing the belief that the most profound knowledge comes from learning through experiences and finding positivity in every setback. I extend my sincere appreciation to my dedicated supervisor, Mr Mohd Khanapiah Bin Nor, whose guidance and insightful ideas played a pivotal role in the successful completion of this project. I am immensely grateful for her expertise and encouragement throughout this journey. Special thanks tomy friends and all those who guided the project's completion. Your support and camaraderie have been invaluable. I also acknowledge the person who ignited my passion for electronic engineering and those who constantly inspired me to overcome obstacles in pursuing this field. To everyone who believed in me,thank you for being adriving force behind my academic achievements. This project is a testament to the collective encouragement and inspiration. I have received from my loved ones and mentors.

ABSTRACT

The rapid proliferation of technology has paved the way for innovative healthcare solutions, enabling individuals to take an active role in monitoring their well-being. This thesis presents the development of an automatic health monitoring system using Arduino, a versatile and cost-effective microcontroller platform. The proposed system aims to provide a convenient and accessible means for individuals to monitor their vital signs, facilitating proactive health management and early detection of potential health issues. The system integrates various biosensors, including temperature and heart rate sensors, to accurately measure and record critical health parameters. The Arduino board serves as the central processing unit, responsible for acquiring data from the sensors, performing necessary computations, and displaying the results on a user-friendly interface. Through extensive research and experimentation, the project explores the implementation of data filtering and smoothing techniques to enhance the accuracy and reliability of sensor readings. Additionally, the system incorporates data logging capabilities, allowing users to maintain a comprehensive record of their health data for future reference and analysis. The development process encompasses hardware and software implementation, rigorous testing, and optimization to ensure the system's functionality and performance. The cost-effectiveness, portability, and user-friendly nature of the developed system make it an attractive solution for individuals seeking to monitor their health proactively, enabling early intervention and informed decision-making regarding their well-being.

ABSTRAK

Penyebaran teknologi yang pesat telah membuka jalan bagi penyelesaian penjagaan kesihatan yang inovatif, membolehkan individu memainkan peranan aktif dalam memantau kesejahteraan mereka. Tesis ini membentangkan pembangunan sistem pemantauan kesihatan automatik menggunakan Arduino, platform mikropengawal yang serba boleh dan berkos efektif. Sistem yang dicadangkan bertujuan untuk menyediakan cara yang mudah dan boleh diakses bagi individu memantau tanda-tanda hayat mereka, memudahkan pengurusan kesihatan proaktif dan pengesanan awal isu kesihatan berpotensi. Sistem ini mengintegrasikan pelbagai penderia bio, termasuk penderia suhu dan kadar denyutan jantung, untuk mengukur dan merekodkan parameter kesihatan penting dengan tepat. Papan Arduino berfungsi sebagai unit pemprosesan pusat, bertanggungjawab memperoleh data daripada penderia, melakukan pengiraan yang diperlukan, dan memaparkan keputusan pada antara muka mesra pengguna. Melalui penyelidikan dan eksperimen yang mendalam, projek ini meneroka pelaksanaan teknik penurasan dan pelicingan data untuk meningkatkan ketepatan dan kebolehpercayaan bacaan penderia. Tambahan pula, sistem ini menggabungkan keupayaan pengekalan log data, membolehkan pengguna mengekalkan rekod data kesihatan yang komprehensif untuk rujukan dan analisis masa depan. Proses pembangunan merangkumi perlaksanaan perkakasan dan perisian, pengujian yang ketat, dan pengoptimuman untuk memastikan kefungsian dan prestasi sistem. Sistem yang dibangunkan yang berkos efektif, mudah alih, dan mesra pengguna menjadikannya penyelesaian yang menarik bagi individu yang ingin memantau kesihatan mereka secara proaktif, membolehkan campur tangan awal dan membuat keputusan berdasarkan maklumat mengenai kesejahteraan mereka.

ACKNOWLEDGEMENTS

I want to begin by conveying my sincere thanks to my supervisor, Mr. Mohd Khanapiah Bin Nor, for his invaluable guidance, wisdom, and steadfast support throughout this project. His expertise and patience have played a pivotal role in shaping the success of my study.

I am profoundly grateful to my parents, and cherished family members fortheir continuous love, encouragement, and prayers throughout my entire period of study. Their consistent support has been a motivating force, and I am genuinely thankful for theirpresence in my life.

Additionally, I would like to give a special acknowledgment to my close friends who have been a constant source of encouragement and collaboration throughout this journey. Their unwavering support has been essential, and I deeply appreciate the strong bondwe share.

I also owe gratitude to all those who have offered valuable suggestions and feedback duringthe course of my research. Your input has significantly contributed to the development ofthis study, and I genuinely value your insights.

Lastly, I want to express my heartfelt appreciation to all individuals who have played asignificant role in the completion of this FYP report. While I may not be able to personally mention each one of you, please be aware that your contributions have not gone unnoticed. Thank you all for your support, and I am sincerely appreciative of your presence in my academic journey.

TABLE OF CONTENTS

		PAGE
DECLARATION		
APPROVAL		
DEDICATIONS		
ABSTRACT		**i**
ABSTRAK		**ii**
ACKNOWLEDGEMENTS		**iii**
TABLE OF CONTENTS		**iv**
LIST OF TABLES		**vii**
LIST OF FIGURES		**viii**
LIST OF APPENDICES		**ix**
CHAPTER 1	**INTRODUCTION**	**1**
1.1	Background	1
1.2	Addressing Global Issue Through Automatic Health Monitoring System Project	3
1.3	Problem Statement	3
1.4	Project Objective	5
1.5	Scope of Project	5
CHAPTER 2	**LITERATURE REVIEW**	7
2.1	Introduction	7
2.2	Understanding Global/Current Issue in the Literature Review	7
2.3	Needs for Monitoring System for the Manufacturing Process	9
2.4	Previous Related Project	9
	2.4.1 Health Monitoring of Human Multiple Physiological Parameters Based on Wireless Remote Medical System	10
	2.4.2 IoT-Based Health Monitoring System Development and Analysis	11
	2.4.3 E-Health System for The Monitoring, Transmission and Storage of The Arterial Pressure of Chronic-Hypertensive Patients	12
	2.4.4 Remote Health Monitoring System for the Estimation of Blood Pressure, Heart Rate, and Blood Oxygen Saturation Level	

2.4.5 Development of smart health monitoring system using Internet of Things 14
2.4.6 IoT and Cloud Based health monitoring system Using Machine learning 15
2.4.7 Investigation into Smart Healthcare Monitoring System in an IoT Environment 15
2.4.8 An IoT Based Secure Patient Health Monitoring System 16
2.4.9 An intelligent health monitoring and diagnosis system based on the internet of things and fuzzy logic for cardiac arrhythmia COVID-19 patients 17
2.4.10 Automatic Wireless Health Monitoring System 18
2.4.11 Development of smart health monitoring system using Internet of Things. 19
2.4.12 An IoT based Patient Health Monitoring System using Arduino Uno 20
2.5 Summary Of Previous Related Project 22

CHAPTER 3 METHODOLOGY 28
3.1 Introduction 28
3.2 Selecting and Evaluating Tools for a Sustainable Development 28
3.3 Project Work Plan 30
3.4 Project Block Diagram 33
3.5 Hardware Specification 35
3.5.1 Arduino Uno 35
3.5.2 MAX30102 – Pulse Oximeter and Heart Rate Sensor 36
3.5.3 LM35 - Temperature Sensor 37
3.5.4 ESP8266 - Wi-Fi Module 37
3.5.5 16x2 LCD Display 38
3.5.6 Resistors 39
3.5.7 LED 40
3.5.8 Donut Board 40
3.5.9 Jumper Wires 41
3.5.10 3.7V Lithium-Ion Battery 42
3.5.11 ECG Sensor 43
3.6 Software Specification 44
3.6.1 Arduino IDE 44
3.6.2 Solidworks 45
3.6.3 Ubidots 45

CHAPTER 4 RESULTS AND DISCUSSIONS 47
4.1 Introduction 47
4.2 System Overview 48
4.3 Experimental Setup 48
4.4 Data Collection and Analysis 48
4.4.1 Heart Rate 48
4.4.2 Oxygen Saturation (SpO2) 50
4.4.3 Body Temperature 51

4.4.4 Electrocardiogram (ECG) 53
4.4.5 ECG Mismatch Between Arduino and Ubidots 53
4.4.5.1 Understanding the Issue 53
4.4.5.2 Causes of Mismatch 53
4.4.5.3 Impact of Missing Data 55
4.5 Threshold and Warning Mechanisms 55
4.6 Data Transmission to Nodemcu 56
4.7 Summary 56

CHAPTER 5 CONCLUSION AND RECOMMENDATIONS 57
5.1 Conclusion 57
5.2 Recommendations 58

REFERENCES 60

APPENDICES 63

LIST OF TABLES

TABLE	TITLE	PAGE
Table 2.1	Summary of previous related project.	22
Table 3.1	Gantt Chart for PSM 1 and PSM 2.	31
Table 3.2	Specification of Arduino uno.	36
Table 3.3	Specification of MAX30102 pulse oximeter and heart rate sensor.	36
Table 3.4	Specification of LM35 sensor.	37
Table 3.5	Specification of ESP8266.	38
Table 3.6	Specification of 16x2 LCD display.	39
Table 3.7	Specification of LED.	40
Table 3.8	Specification of breadboard.	41
Table 3.9	Specification of jumper wires.	42
Table 3.10	Specification of 3.7V lithium-ion battery.	43
Table 4.1	Data collection of three readings for 15 patients for heart rate.	49
Table 4.2	Data collection of three readings for 15 patients for ocygen level.	51
Table 4.3	Data collection of three readings for 15 patients for body temperature.	52

LIST OF FIGURES

FIGURE	TITLE	PAGE
Figure 2.1	Schematic diagram of wireless telemedicine network	11
Figure 2.2	Finger clip used for measure	14
Figure 2.3	Smart healthcare device block diagram	16
Figure 3.1	PSM flowchart	22
Figure 3.2	Project flowchart	33
Figure 3.3	Block diagram of automatic health monitoring system using arduino	34
Figure 3.4	Arduino uno	35
Figure 3.5	MAX30102 pulse oximeter and heart rate sensor	36
Figure 3.6	LM35 sensor	37
Figure 3.7	ESP8266	38
Figure 3.8	16x2 LCD display	39
Figure 3.9	Resistor of 220 ohm	39
Figure 3.10	LED	40
Figure 3.11	Donut board	41
Figure 3.12	Jumper wires	42
Figure 3.13	3.7V lithium-ion battery	43
Figure 3.14	ECG sensor	44
Figure 3.15	Arduino IDE	45
Figure 3.16	Solidworks	45
Figure 3.17	Ubidots	46
Figure 4.1	Data analysis for heart rate	49
Figure 4.2	Data analysis for oxygen level	50
Figure 4.3	Data analysis for body temperature	52

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
Appendix A	Coding for NodeMCU Automatic Health Monitoring System	63
Appendix B	Coding for NodeMCU Automatic Health Monitoring System	70

CHAPTER 1

INTRODUCTION

1.1 Background

Over the past few years, wearable technology and remote monitoring systems have become increasingly popular in the healthcare sector. Real-time monitoring of patients' vital signs and medical data is now possible because to these technologies, which have completely changed the way healthcare is provided. The Automatic Health Monitoring System using Arduino is a pioneering project that aims to address the growing demand for continuous and effortless health monitoring while reducing the burden on healthcare professionals and facilitating early intervention.

Traditional healthcare practices often rely on periodic visits to medical facilities for check-ups and monitoring. However, this approach can be time-consuming, inconvenient for patients, and may fail to detect critical health conditions between visits. Moreover, manual monitoring can be prone to human error and may not provide a comprehensive picture of a patient's health status over time. With the advent of wearable technology and the Internet of Things (IoT), there is a significant opportunity to develop intelligent, automated systems that can continuously track vital signs and medical data without requiring manual intervention.

Today nurses play a critical role in patient monitoring through manual vital sign checks, which is dominant in traditional health care system. This way, it is slow and error prone as well as allows for infrequent monitoring only. On the other hand, currently there is an increased call for a more elaborate treatment program fueled by an increasing number of old people and an increase in chronic diseases’ incidences. As such, this situation necessitates real time monitoring systems, however, due to lack of enough human resource; this remains a hard task to do. Technology, particularly IoT and wearables, offers a promising solution by enabling real-time data collection and remote monitoring. However, many current solutions are prohibitively expensive. Arduino, as a low-cost, open-source microcontroller, provides a solution by enabling real-time data collection and remote monitoring.[1] Recent studies, such as a 2021 Arduino-based ECG monitor, a 2022 IoT-based patient system, and a 2023 low-cost vital sign tracker, have demonstrated Arduino's viability in healthcare. Building on this foundation, our project leverages Arduino's affordability to focus uniquely on enhancing nurse efficiency, aiming to revolutionize ward rounds and address one of healthcare's most pressing challenges.

In the project titled “Development of Automatic Health Monitoring System Using Arduino,” I aim to create an integrated health monitoring system that automatically measures and records various vital signs, including body temperature, heart rate (pulse), oxygen levels (SpO2), and ECG readings. The system will utilize an Arduino microcontroller as its central component, along with sensors such as a pulse sensor, temperature sensor, pulse oximeter (SpO2 sensor), and an ECG module. By connecting to an IoT platform, this system allows remote monitoring of patients’ health, enabling healthcare professionals to access real-time data and take timely actions based on abnormal readings. Ensuring calibration, accuracy, and safety are critical for reliable health measurements in this comprehensive project.

1.2 Addressing Global Issue Through Automatic Health Monitoring System Project

Using technology to address global health issues, the Automatic Health Monitoring System is a large, ambitious initiative. An innovative system that continuously monitors individual health data and vital signs is comprised of a network of wearable devices and networked sensors. This data may be analyzed in real-time by the system by utilizing sophisticated algorithms and machine learning, which allows for the early identification of possible health problems and prompt actions. The system has the potential to improve population health by reducing the effect of diseases and promoting early detection and prevention through its smooth interaction with emergency services and healthcare professionals. Proactive and customized healthcare management made possible by this research has the potential to completely transform healthcare accessible, especially in underserved or remote places. The Automatic Health Monitoring System seeks to enhance preventative care and healthy lifestyles in individuals and, in the process, make a positive impact on the global society by providing them with actionable information into their well-being.

1.3 Problem Statement

In modern healthcare settings, nurses manually monitor and record each patient's vital signs during a significant portion of their shifts during routine ward rounds. This repetitious task wastes time that could be better used on critical care tasks including medicine administration, wound care, and patient education and support when there are high patient-to-nurse ratios. Because these rounds take a lot of time, nurses who perform them not only become more stressed and tired, which raises the risk of burnout, but they also delay responding to crises because they may miss vital signs changes. In addition, errors can occur during human data entry, and there may be irregular intervals between monitoring. We suggest an Arduino-based Automatic Patient Health Monitoring System to address this urgent problem. Our device automates routine checkups by continually and correctly measuring vital signs such as oxygen levels, temperature, heart rate, and ECG. It also transmits real-time data to a central server and sends automated alarms for abnormal readings. By considerably lowering the amount of time nurses must spend on ward rounds, this solution enables them to concentrate on providing critical care, which enhances patient safety, improves outcomes, and lessens nurse burnout. In order to allow patients to accurately and continuously monitor their vital signs in real-time, an autonomous health monitoring system that is portable, reasonably priced, and simple to use is therefore required. The suggested solution attempts to address these issues by creating an automated health monitoring system that measures and displays vital signs effectively and precisely utilizing an Arduino microcontroller board, which is inexpensive. A number of sensors will be used in the system to measure the necessary parameters, and an LCD display will present the findings. The correctness, dependability, and usability of the suggested system will be assessed, and its possible influence on enhancing healthcare services will be explored.

1.4 Project Objective

The main objective of the thesis project titled "Development of Automatic Health Monitoring System Using Arduino to design and implement a low-cost, user-friendly, and accurate health monitoring system that can measure vital signs such as body temperature, heart rate, oxygen levels, and ECG readings using Arduino-based sensors. The system aims to provide real-time monitoring and display of these vital signs on an LCD screen, Facilitating better health condition monitoring and management for both patients and healthcare providers. The project also aims to assess the produced system's performance in terms of dependability, correctness, and usability. The specific objectives of the project include:

a) Designing and implementing a system architecture for the automatic health monitoring system using Arduino.

b) Integrating various sensors such as temperature sensors, pulse sensors, oxygen level sensors, and ECG sensors with the Arduino board to measure the required vital signs.

c) Evaluating the correctness, dependability, and user-friendliness of the developed system.

The project aims to contribute to the development of low-cost and accessible health monitoring systems that can improve healthcare outcomes, especially in resource-limited settings.

1.5 Scope of Project

The scope of this project is to develop an automatic health monitoring system using Arduino to measure vital signs such as body temperature, heart rate, oxygen levels, and ECG readings. The project aims to design and implement a low-cost, user-friendly, and accurate health monitoring system that can provide real-time monitoring and display of these vital signs on an LCD screen. The system will integrate various sensors with the Arduino board to measure the required vital signs and calibrate the sensors to ensure accurate and reliable measurements. The project will also involve developing algorithms for processing and analyzing the measured data to extract relevant information such as heart rate variability, and oxygen saturation levels. The performance of the developed system will be evaluated in terms of accuracy, reliability, and ease of use, and the results will be compared with existing commercial health monitoring systems. The project will contribute to the development of low-cost and accessible health monitoring systems that can improve healthcare outcomes, especially in resource-limited settings.The project will be divided into several phases, including:

Literature review and system design: This phase will involve reviewing existing literature on health monitoring systems and designing a system architecture for the proposed system using Arduino.

Sensor selection and integration: This phase will involve selecting appropriate sensors for measuring the required vital signs and integrating them with the Arduino board.

System implementation and testing: This phase will involve implementing the system and testing it to ensure that it meets the design specifications.

Performance evaluation: This phase will involve evaluating the performance of the developed system in terms of accuracy, reliability, and ease of use.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Given its potential to completely transform the way healthcare is delivered, the creation of an Arduino-based automated health monitoring system has attracted a lot of interest recently. The goal of the system is to make it simpler for patients and healthcare professionals to monitor and manage health concerns by providing real-time monitoring and presentation of vital signs such oxygen levels, body temperature, heart rate, and ECG readings. An overview of current health monitoring systems, Arduino applications for health monitoring, and the integration of many sensors for vital sign measurement are given in this review of the literature. The design and implementation of comparable systems, techniques for evaluating performance, and possible uses of these systems in healthcare contexts are also included in this paper. The literature analysis indicates that the creation of an accurate, low-cost, and user-friendly automatic health monitoring system with Arduino has the potential to overcome the drawbacks of current health monitoring systems. To assess the accuracy, dependability, and use of such systems, more investigation is necessary.

2.2 Understanding Global/Current Issue in the Literature Review

The development of an autonomous health monitoring system using Arduino is motivated by several global and current issues in healthcare. Firstly, the growing prevalence of chronic diseases and the geriatric population have placed significant strain on traditional healthcare systems. According to the World Health Organization (WHO), Cardiovascular diseases account for most NCD deaths, or at least 19 million deaths in 2021, followed by cancers (10 million), chronic respiratory diseases (4 million), and diabetes (over 2 million including kidney disease deaths caused by diabetes)[1]. Monitoring vital signs and early detection of health issues can play a crucial role in managing these chronic conditions and enhancing patient outcomes.

Additionally, the COVID-19 pandemic has highlighted the importance of remote monitoring and telemedicine solutions. The need for social distancing and minimizing hospital visits has accelerated the adoption of remote patient monitoring technologies [2]. An automatic health monitoring system using Arduino can provide a cost-effective and accessible solution for individuals to monitor their health from the convenience of their residences, reducing the risk of exposure and alleviating the burden on healthcare facilities.

Furthermore, the advent of the Internet of Things (IoT) and wearable technologies has opened up new possibilities for continuous and non-invasive health monitoring. Integrating Arduino-based systems with these technologies can facilitate real-time data collection and analysis, empowering individuals to take a proactive approach to their health and well-being [3].

Moreover, access to affordable and reliable healthcare remains a challenge in many regions of the globe, particularly in resource-constrained settings. The development of low-cost and open-source solutions like Arduino-based health monitoring systems can contribute to bridging this disparity and promoting health equity [4].

By addressing these global and current issues, the development of an autonomous health monitoring system using Arduino has the potential to revolutionize healthcare delivery, enhance patient outcomes, and promote preventive care. The literature review emphasizes the imperative need for such solutions and the potential impact they can have on individual and public health.

2.3 Needs for Monitoring System for the Manufacturing Process

The modern instruments are needed to support the current modern lifestyle, which has become a requirement in and of itself. The monitoring system for the manufacturing process has consequently turned into a problem for society in the industrial sector. This system is relatively new to the market, and not many businesses employ it for their particular industries. Consequently, this monitoring system for the production process can aid in time and energy savings. This can be accomplished by utilising this monitoring system for the manufacturing process, which will enable anybody, anywhere to view the machine's real-time data.

2.4 Previous Related Project

Previous related work describes a researcher who carried out a planned project in a comparable manner. Numerous researchers worldwide have conducted nearly identical studies; however, their tools and approaches for carrying out the research varied. I choose twelve of the most comparable articles and provide a summary in order to finish this section. All of the selected articles are mentioned in the reference.

2.4.1 Health Monitoring of Human Multiple Physiological Parameters Based on Wireless Remote Medical System

Traditional medical technology requires patients to visit hospitals, which is inconvenient for those with irregular or sudden illnesses and those needing periodic care. Telemedicine addresses this by enabling real-time remote monitoring of patients' physiological conditions, thus mitigating the impact of sudden illnesses and controlling infectious diseases [6]. In the U.S., telemedicine leverages satellite technology, especially benefiting remote areas like Alaska. Europe, with its advanced mobile communication networks, uses these for telemedicine, effectively implemented in various countries. Other regions, including South Africa, Japan, and Australia, have also developed telemedicine systems suited to their needs. Greece, for example, uses mobile networks to connect ambulance systems with hospital monitoring centers. Technological advancements, such as Wi-Fi and multi-network fusion, enhance telemedicine's capabilities. Home monitoring systems and virtual medical data instruments further facilitate remote care. This paper presents a wireless remote medical health monitoring system using sensor technology, detailing algorithms for processing respiratory, pulse, and ECG data. The system was thoroughly tested, verifying the accuracy of these physiological signals and calculation algorithms. The study is organized into sections covering telemedicine system research, signal processing algorithms, system testing, and future research directions.

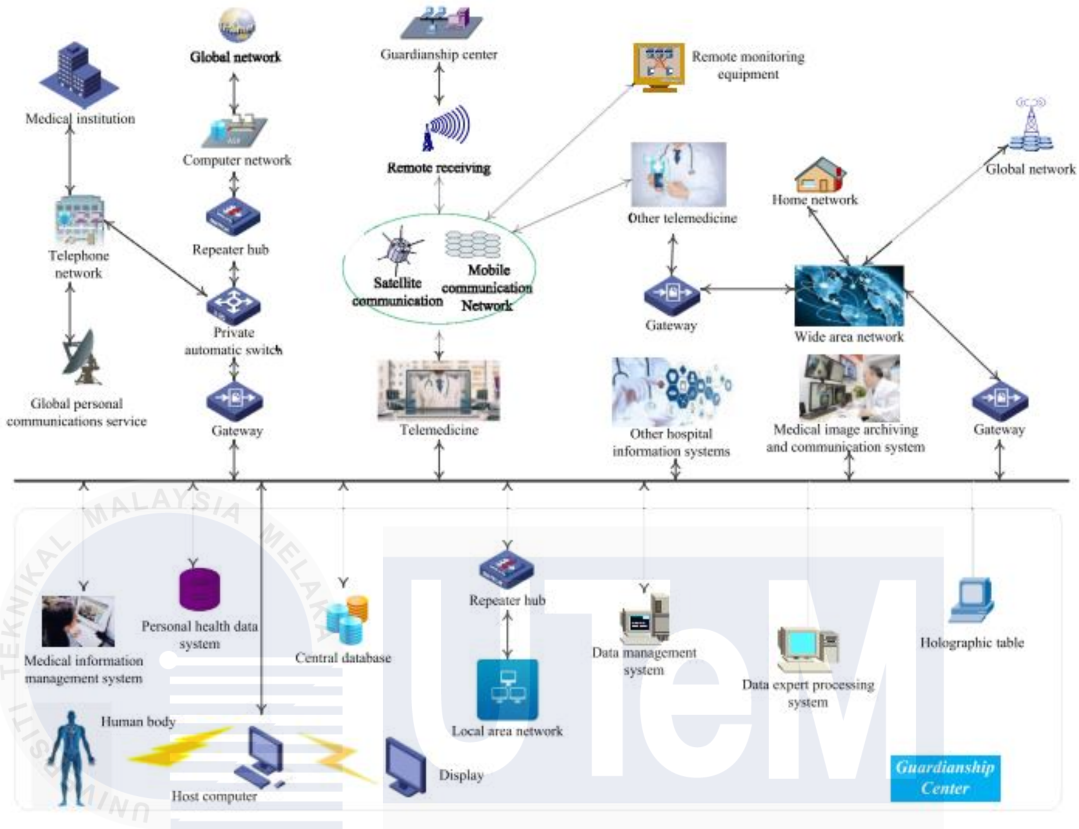

Figure 2.1 Schematic diagram of wireless telemedicine network. [6]

2.4.2 IoT-Based Health Monitoring System Development and Analysis

The design and execution of an Arduino-based Internet of Things health monitoring system are presented in this journal article. The system uses sensors such as the MAX30100 and LM35 to assess a patient's blood oxygen saturation (SpO2) levels, heart rate, and body temperature. The collected data is transmitted wirelessly to a mobile application via Bluetooth, allowing patients to monitor their health remotely. Aimed at providing an affordable and accessible solution for people in rural areas of developing countries like Bangladesh, where access to healthcare facilities is limited, the system leverages an Arduino Uno microcontroller board integrated with sensors [7]. The hardware components also include a Bluetooth module and an LCD display, while the software development utilizes the MIT App Inventor. Real-life testing was conducted on three human subjects, and the results for body temperature, heart rate, and SpO2 measurements are presented and analyzed. The authors highlight the potential benefits of enabling continuous health monitoring, early detection of abnormalities, and facilitating timely medical interventions. Future improvements suggested include integrating additional sensors, enhancing security algorithms, and migrating to a Raspberry Pi platform.

2.4.3 E-Health System for The Monitoring, Transmission and Storage of The Arterial Pressure of Chronic-Hypertensive Patients

The journal article presents an e-Health prototype system for the monitoring, transmission, and storage of arterial blood pressure data of chronic hypertensive patients. The system utilizes a wrist-worn digital blood pressure monitor to measure the patient's systolic and diastolic blood pressure, as well as heart rate [8]. The collected data is transmitted via Bluetooth to an Arduino board, which then sends it in real-time to a remote server through a cellular network (GSM/GPRS). The data is stored in a database and can be accessed by the patient and doctor through a web interface, allowing for remote monitoring and tracking of the patient's blood pressure history. Notably, the system incorporates an alert mechanism that sends an SMS notification to a predefined number if the patient's blood pressure readings exceed certain thresholds, indicating a potential emergency situation. The prototype was validated on real patients, and the blood pressure measurements were compared with a WHO-certified device, showing acceptable variations within ±5 points. The authors highlight the benefits of the system, such as enabling timely monitoring and intervention, reducing the need for frequent hospital visits, and optimizing healthcare resources. Future improvements are suggested, including exploring other IoT protocols, incorporating additional biomedical data sensors, and adapting the web interface for mobile devices.

2.4.4 Remote Health Monitoring System for the Estimation of Blood Pressure, Heart Rate, and Blood Oxygen Saturation Level

This work designs and implements a remote health monitoring system based on the Internet of Things (IoT) to measure blood pressure (BP), heart rate (HR), and blood oxygen saturation levels (SpO2). Utilizing a single photoplethysmography (PPG) signal, the developed sensor is a wearable ring-shaped device that measures and extracts the PPG signal from the finger. The largest peak of the PPG signal, which equates to the BP values, was found using an automatic multiscale-based peak (AMBP) detection technique. The sensor was interfaced with an Arduino 1010 WIFI MKR for remote health monitoring [9].

A Contec ambulatory blood pressure sensor and a Braun pulse oximeter monitor were used as benchmarks to compare the new sensor to two common commercial measurement tools. A graphical representation was created using the MATLAB ThingSpeak platform, and the estimated BP, HR, and SpO2 readings were tracked remotely. Comparing the created sensor's accuracy to commercial devices, it was determined to be within acceptable comparable limits. The authors talk about how their technology could be useful for ongoing remote health monitoring and early clinical deterioration detection.

Figure 2.2 Finger clip used for measure.

2.4.5 Development of smart health monitoring system using Internet of Things

The author presents a study on the development of a smart health monitoring system using Internet of Things technology. The system aims to automate the measurement of various health parameters of patients, providing accurate and efficient monitoring [10]. It discusses the use of sensors, Node MCU, GSM technology, and other components to collect and transmit data for analysis. The study highlights the benefits of automation in the medical field, emphasizing the importance of continuous monitoring and remote access to health data for healthcare professionals. The research also discusses upcoming projects that will use artificial intelligence and machine learning algorithms to generate prescriptions and diagnoses automatically.

2.4.6 IoT and Cloud Based health monitoring system Using Machine learning

In this project, the creation of an Internet of Things (IoT)-based smart health monitoring system is covered. The system, designed by M. Pravin Savaridass et al., attempts to continually monitor a number of patient health factors, including oxygen level, temperature, pressure, pulse rate, and glucose level. Sensors such as the LM35 for temperature, the pressure sensor for pressure monitoring, the heartbeat sensor for pulse rate, the pulse oximeter for oxygen content, and the glucometer for blood glucose level are important parts. Healthcare providers can easily access data transmitted to Google Sheets and the cloud thanks to the Node-MCU microcontroller and Wi-Fi module. When measured values diverge, an email or SMS alert is sent out. Benefits of the system include enhanced accuracy, decreased manual labor, remote monitoring, and better patient quality of life. Future research will integrate artificial intelligence and machine learning algorithms to provide prescriptions and diagnoses automatically. Information about comparable IoT-based health monitoring systems and their advantages can be found in a number of references. In the comparative study of the models that were used by various authors for health monitoring systems where the SVM attained the highest accuracy of 99.1 percent after testing the suggested model, which is promising for purposes [11].

2.4.7 Investigation into Smart Healthcare Monitoring System in an IoT Environment

The creation of a Smart Healthcare Monitoring System with the Internet of Things and Raspberry Pi is covered in the journal. Patient data is collected via the system's temperature, blood pressure, ECG, and pulse rate sensors. In order to monitor and prescribe treatments, doctors may access this data on an Internet of Things server. When sensor values above predetermined thresholds, an alert is produced. The system communicates with data and sends alerts using Python, RASPBIAN OS, embedded C, and Wiring-Pi. Sensor values are stored in the cloud, giving patients and physicians remote access. It was carried out by Salini Pradhan et al.

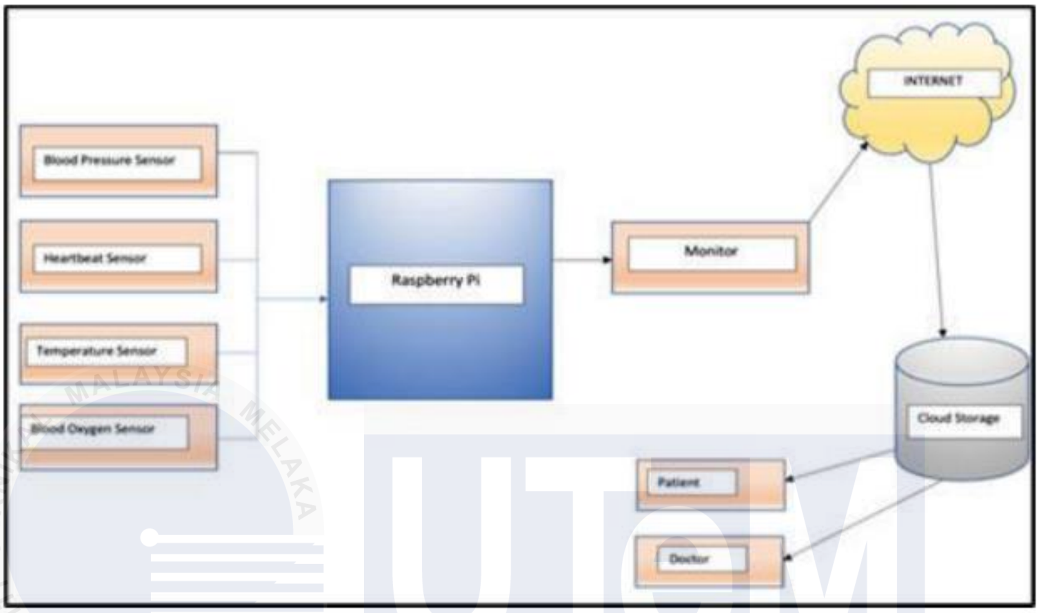

Figure 2.3 Smart healthcare device block diagram[12]

2.4.8 An IoT Based Secure Patient Health Monitoring System

The project discusses how machine learning is being used to develop a cloud-based and Internet of Things-based health monitoring system. The authors, Chhavi Rana, an associate professor in the same field, and Preeti, a research scholar in computer science and engineering at Maharishi Dayanand University in Rohtak, Haryana, India, emphasize the value of in-home health care services, which let patients receive medical care in the comfort of their own homes. Patients in remote areas can communicate with doctors in cities with this technology to facilitate remote health monitoring. Vital signs including blood pressure, temperature, pulse, and heart rate are tracked via wearable sensors[13]. Algorithms based on machine learning are used to forecast illnesses and identify physicians for consultation. Real-time monitoring of patients' vital signs is made possible by the combination of IoT technology and cloud-based solutions, which promotes individualized and responsive healthcare. The authors emphasize the need of reliable, affordable, and safe health monitoring systems in order to successfully prevent and treat chronic, high-risk diseases. They also recommend more study to improve IoT-based health monitoring systems' capacity for risk prediction.

2.4.9 An intelligent health monitoring and diagnosis system based on the internet of things and fuzzy logic for cardiac arrhythmia COVID-19 patients

The paper "An intelligent health monitoring and diagnosis system based on the internet of things and fuzzy logic for cardiac arrhythmia COVID-19 patients" [10]provides a thorough analysis of a state-of-the-art system intended to track and identify cardiac arrhythmia in COVID-19 patients. This study's author is M.Z. Rahman et al, which the objective is to design and implement an intelligent health monitoring and diagnosis system for critical cardiac arrhythmia COVID-19 patients[14].

The system analyzes ECG readings and provides precise diagnoses by utilizing fuzzy logic, artificial intelligence, and the internet of things. The system is capable of classifying various serious arrhythmia states by picking particular characteristics from the original signals and filter banks. The study highlights how crucial it is to choose features wisely in order to save computing power without sacrificing accuracy.

In addition, the system makes use of defuzzification methods and fuzzy membership functions to manage language variations and membership levels efficiently. Processing ECG readings, creating fuzzy rules based on statistical indications, and optimizing the system's performance are all part of the process.

The MIT-BIH database and Internet of Things technologies are also included in the research as ways to improve COVID-19 patient monitoring. The outcomes of the trials show how accurately the suggested intelligent system can diagnose cardiac arrhythmias.

Overall, the study shows how incorporating fuzzy logic and the Internet of Things into healthcare systems might enhance critical condition monitoring and diagnosis for COVID-19 patients.

2.4.10 Automatic Wireless Health Monitoring System

Vikramsingh R. Parihar's work, which used the ATmega328 microcontroller (Arduino Uno) to construct a wireless heartbeat and temperature monitoring system, is presented in the journal named "Automatic Wireless Health Monitoring System". The principal aim of the system is to provide remote, real-time patient monitoring, therefore meeting the demand for ongoing health surveillance. The device measures body temperature and heartbeat via sensors, providing essential physiological data that is easily shown on an LCD panel. The project consists of sensors which measures heartbeat and body temperature of a patient which is controlled by the microcontroller and both the readings are displayed on the LCD monitor [15].

The system consists of a transmitter portion and a receiver section. Sensors in the transmitter section measure the patient's vital signs and communicate the data to the microcontroller for wireless communication. The patient's temperature and heartbeat are shown on an LCD screen at the receiving end once the sent data is decoded using a nRF module. This novel strategy emphasizes the value of real-time remote monitoring in healthcare and has the potential to improve patient care and early health issue diagnosis.

The PDF also includes references to other relevant publications in the subject of health monitoring systems, displaying a range of technologies and approaches applied to comparable goals. Researchers and medical practitioners may learn a great deal about the changing automatic wireless health monitoring system market and how these developments may affect patient outcomes and healthcare delivery.

2.4.11 Development of smart health monitoring system using Internet of Things.

An in-depth analysis of an Internet of Things (IoT)-based smart health monitoring system is provided in this publication. The research, written by M. Pravin Savaridass and associates, explores the development of an affordable and readily deployable system intended to monitor a number of vital patient health indices. Development in technologies has made human life much easier and so much simple [16]. The system makes it possible to measure vital indicators such blood pressure, blood glucose, heart rate, and temperature continuously and accurately by utilizing sensors, Node MCU, and other components.

The suggested approach has a number of benefits, such as improved cost effectiveness, accurate data collection, less manual labor for medical staff, and the ability to send patient data to experts for additional review with ease. The system guarantees real-time monitoring and analysis of patient health parameters by integrating IoT platforms, which enables timely medical treatments when needed.

The article also describes the system's potential for the future and includes plans to integrate artificial intelligence and machine learning technologies. By automating the diagnostic procedure, improving reporting analysis, and producing prescriptions, these developments seek to further maximize the efficacy and efficiency of the health monitoring system. All things considered, M. Pravin Savaridass and his colleagues' work highlights how IoT-based solutions have the ability to transform healthcare administration and enhance patient outcomes.

2.4.12 An IoT based Patient Health Monitoring System using Arduino Uno

The PDF created by V. Akhila and colleagues explores the design and execution of a cutting-edge Arduino Uno-based Internet of Things patient health monitoring system. By using smart sensors to record vital signs like body temperature and heartbeat [18zhang], the device is intended to transform healthcare. After Arduino microcontrollers have processed the data from various sensors, it is wirelessly sent to an Internet of Things platform for in-depth analysis and visualization. This configuration gives medical personnel access to real-time patient data, allowing them to act quickly and decisively in life-threatening circumstances.

The authors highlight how IoT technology has the ability to completely change the healthcare industry. They stress how it can improve patient care, facilitate proactive healthcare management, and maximize communication between patients and healthcare professionals. The proposed Patient Health Monitoring System seeks to improve patient outcomes, the quality of healthcare delivery, and the establishment of a more responsive and effective healthcare ecosystem by utilizing the capabilities of IoT devices and Arduino technology.

The study by V. Akhila et al. emphasizes how critical it is to use IoT developments to develop intelligent healthcare solutions that put patients' needs first and allow for individualized, data-driven treatment. The authors' study promotes the use of IoT-based systems in healthcare settings to provide patients and healthcare providers with useful information and resources for proactive health management.

2.1 Summary Of Previous Related Project

Table 2.5 Summary of Previous Related Project

No.	Author	Title	Year	Description
1.	Zhang K, Ling W	Health Monitoring of Human Multiple Physiological Parameters Based on Wireless Remote Medical System	2020	Telemedicine is emerging as a transformative model that enables remote information sharing and monitoring, ultimately ensuring that everyone has equal access to healthcare resources. Building on this vision, our research proposes a multi-parameter wireless health monitoring system. It accurately tracks a comprehensive range of vital signs—temperature, respiration, blood oxygen, pulse, blood pressure, and ECG—using innovative techniques. These include fabric electrodes for bioimpedance, Fourier transform for respiratory signals, photoelectric sensors for pulse, cuff-free blood pressure measurement via pulse wave timing, near-infrared technology for SpO2, and QRS band extraction for ECG. Rigorous testing has proven the validity and reliability of our system's data, marking a significant step toward more accessible, high-quality remote healthcare. [6]
2.	Khan M, Alanazi T, Albraikan A, Almalki F	IoT-Based Health Monitoring System Development and Analysis	2022	In today's innovative landscape, this paper introduces an IoT-based health monitoring system designed to make medical device usage simpler and more affordable. Particularly relevant in developing countries like Bangladesh, where rural patients often lack access to clinics, this system offers a cost-effective alternative to expensive hospital visits. Using a three-layer architecture—physical, logical, and application—it accurately measures body temperature, heartbeat, and blood oxygen levels, then transmits this data via Bluetooth to a custom MIT app. With

				a 95% confidence interval and 5% maximum error, it ensures reliable readings. By enabling at-home monitoring, this system not only increases affordability but also enhances personal healthcare accessibility. In critical times, such as during the COVID-19 pandemic, this IoT solution could significantly impact lives by providing essential health data to those who need it most.[7]
3.	Lopez A, Jimenez Y, Bareno R, Balamba B, Sacristan J	E-Health System for The Monitoring, Transmission and Storage of The Arterial Pressure of Chronic-Hypertensive Patients	2019	New advances in technology facilitate access to health services from anywhere. This research focuses on patients with high blood pressure (HTA) who must periodically travel far for check-ups. The article presents an e-Health prototype that measures systolic pressure, diastolic pressure, and heart rate, transmitting data via Bluetooth to a processing card and in real time to a remote server through the cellular network. Data is stored in a database, allowing patients and doctors to monitor these variables from any computer. In case of abnormalities, the system sends alerts via text message. The prototype, validated with highly acceptable results in several patients, was compared to a World Health Organization-certified blood pressure monitor, showing a variation of ±5 points. Real-time sending takes less than 23 seconds from sensor to prototype, while web visualization takes under two minutes, depending on coverage. Given its reliability and usability, the system significantly improves timely data collection, helping prevent potentially fatal emergencies.[8]
4.	Nwibor C, Haxha S, Ali M, Sakel M, Haxha A, Saunders K, Nabakooza S	Remote Health Monitoring System for the Estimation of Blood Pressure, Heart Rate, and Blood	2023	This article presents an Internet of Things (IoT)-based remote health monitoring system design and implementation for estimating blood pressure (BP), heart rate (HR), and blood oxygen saturation levels (SpO2). The device, using a biomedical sensor with an embedded signal condition unit, collects, evaluates, predicts, and reads health data from a single

		Oxygen Saturation Level		photoplethysmography (PPG) signal. Data is stored on the 'ThinkSpeak' IoT platform and displayed locally on a 0.91 organic light-emitting diodes (OLEDs) screen. A computer-based algorithm uses selected parameters from the PPG signal to estimate BP (systolic and diastolic), HR, and SpO2. An automatic multiscale-based peak (AMBP) detection algorithm identifies the PPG signal's maximum peak. The developed sensor, worn as a ring and interfaced with an Arduino 1010 WIFI MKR, was benchmarked against standard devices: a Contec ambulatory BP sensor and a Braun pulse oximeter monitor. The estimated values are remotely monitored and graphically represented.[9]
5.	Savaridass M, Ikram N, Deepika R, Aarnika R	Development of smart health monitoring system using Internet of Things	2021	Here's a concise summary of the provided text in paragraph form, preserving the original terminology: "In this modern world, automation plays a vital role, particularly in the medical field, reducing human work and providing more benefits with greater accuracy than manual methods. The patient's health monitoring system is one such automated system, measuring various health-related physical parameters. The medical field is growing by using IoT platforms, which help to get patient information quickly. Basic physical parameters measured are temperature (using LM35 sensor), pressure (using digital sphygmomanometer), pulse rate (using LM358 OPAMP heartbeat sensor), oxygen content in haemoglobins (using pulse oximetry), and blood glucose level (using glucometer). Node-MCU serves as the microcontroller, its Wi-Fi module sending measured physical parameters to the cloud and Google sheets. If any discrepancy is detected in the measured values, an alert message (E-mail or SMS) is sent to the doctor and the patient's relative.[10]

6.	Preeti, Rana C	IoT and Cloud Based health monitoring system Using Machine learning	2022	The health care sector is focusing on in-home services, allowing patients to receive care privately at home. A patient in a rural region can use a remote health monitoring system to communicate with a doctor in a larger city. Machine learning has been used for smart health monitoring systems, employing a wearable sensor to identify five parameters: Electrocardiogram (ECG), pulse rate, pressure, temperature, and position detection. The technology uses machine learning algorithms to identify doctors for consultation and predict ailments. Coupling IoT technology with health monitoring provides more personalized and responsive care, primarily monitoring patients' vital signs in real-time. Authorized individuals can access these signs from their smartphone or PC via a cloud server. After testing the suggested model, the Decision Tree (DT) attained the best accuracy of 99.1 percent, which is promising. It is observed that the DT achieves best accuracy, while Random Forest is the second-best classifier for this problem.[11]
7.	Salini Pradhan, Ahmad Anwar Zainuddin, Rohilah Sahak, Mohamad Yunos M	Investigation into Smart Healthcare Monitoring System in an IoT Environment	2022	The health care sector is enhancing in-home services, allowing patients to receive medical care at home. Patients in rural areas can use remote monitoring systems to communicate with doctors in cities. Machine learning, using wearable sensors to track ECG, pulse rate, pressure, temperature, and position, helps identify doctors for consultation and predict ailments. By combining IoT and health monitoring, the system offers personalized, real-time monitoring of patients' vital signs, accessible via cloud servers on smartphones or PCs. The Decision Tree algorithm achieved the highest accuracy of 99.1 percent, with Random Forest as the second-best classifier.[12]
8.	Yadav K, Alharbi A, Jain A, Ramadan R	An IoT based secure patient health monitoring system	2022	The rapid growth of artificial intelligence and communication technologies has led to the emergence of the Internet of Things (IoT) field. IoT technology in modern healthcare provides real-

				time monitoring, proper administration of patient information, and effective healthcare management. However, if patient information is not securely maintained during transfer or storage, IoT usage can become problematic. This manuscript presents a secure IoT healthcare monitoring system using the Blockchain-based XOR Elliptic Curve Cryptography (BC-XORECC) technique to prevent various attacks. The system first establishes an authentication process for patient details, generating tokens, keys, and tags using Length Ceaser Cipher-based Pearson Hashing Algorithm (LCC-PHA), Elliptic Curve Cryptography (ECC), and Fishers Yates Shuffled Based Adelson-Velskii and Landis (FYS-AVL) tree to prevent unauthorized access. Secure data transfer is then performed using BC-XORECC, which ensures high data privacy and blocks attackers. Finally, the Linear Spline Kernel-Based Recurrent Neural Network (LSK-RNN) classifies the patient's health status. This framework ensures secure data transfer without data loss or breaches and is efficient for healthcare monitoring via IoT. Experimental analysis shows that the proposed framework achieves faster encryption and decryption times, classifies health status with 89% accuracy, and is robust compared to existing methods[13].
9.	Rahman M, Akbar M, Leiva V, Tahir A, Riaz M, Martin-Barreiro C	An intelligent health monitoring and diagnosis system based on the internet of things and fuzzy logic for cardiac arrhythmia COVID-19 patients.	2023	During the COVID-19 pandemic, there is a need for intelligent health systems for critical heart patients. This study designs an AI-based system with IoT monitoring and fuzzy logic diagnosis, enabling remote care for isolated patients. Using sensors, cloud storage, and communication tools, the system processes ECG signals through digital filters for feature assessment. Trained on MIT-BIH data, the system achieves nearly 100% accuracy, making it particularly beneficial for isolated COVID-19 patients with critical heart arrhythmias[14].

10.	Vikramsingh R Parihar	Automatic Wireless Health Monitoring System	2018	This project describes the development of a wireless system for remote real-time monitoring of a patient's heartbeat and body temperature. It uses an ATmega 328 microcontroller (Arduino Uno) and sensors to measure heartbeat and temperature. The readings are displayed on an LCD monitor. A wireless system transmits the measured data to a remote location, where it is displayed on another LCD. The heartbeat sensor counts the beats per minute, while the temperature sensor measures the body temperature. Both readings are sent to the microcontroller for transmission to the receiving end.[15]
11.	M. Pravin Savaridass, N. Ikram, R. Deepika, R. Aarnika	Development of smart health monitoring system using Internet of Things.	2019	Automation is vital in modern medicine, providing accurate patient health monitoring. This system uses sensors to measure temperature (LM35), blood pressure, pulse rate (LM358 OPAMP), oxygen levels (pulse oximetry), and blood glucose. A Node-MCU microcontroller with Wi-Fi sends data to the cloud and Google Sheets. Abnormal readings trigger alerts to doctors and relatives, leveraging IoT for rapid information access.[16]
12.	V.Akhila, Y.Vasavi, K.Nissie, P.Venkat Rao	An IoT based Patient Health Monitoring System using Arduino Uno	2017	IoT connects physical objects through the internet, making users' lives more comfortable. IoT devices collect data like temperature, blood pressure, and sugar levels to evaluate patient health. This project proposes an IoT-based Patient Health Monitoring System (PHMS) using Arduino to collect required parameters from sensor devices. PHMS with Arduino notifies patients with precautions and suggests medical care in critical situations. The Arduino Uno board collects sensor data and sends it wirelessly to an IoT website. PHMS evaluates parameters like heartbeat, temperature, blood pressure, enabling data-driven decisions.[17]

CHAPTER 3

METHODOLOGY

3.1 Introduction

The project's methodology will be described and explained in this chapter. It will be carried out in a way that comprehends and satisfies the project analysis in order to accomplish the intended goals. There are three parts to this chapter. The project flow was the first component, and the block diagram—which can build a project from top to bottom—came next. Finally, this chapter covered every hardware and software component that was required for this project.

3.2 Selecting and Evaluating Tools for a Sustainable Development

The sustainable development of an autonomous health monitoring system using Arduino requires careful selection and evaluation of the appropriate tools and technologies. To ensure long-term viability and adaptability, the following factors should be considered:

a) Open-source hardware and software: Arduino is an open-source platform, which offers several advantages for sustainable development. It promotes transparency, collaborative development, and community support, ensuring the system's longevity and continuous refinement. Additionally, open-source tools foster cost-effectiveness and accessibility, aligning with the aim of creating affordable healthcare solutions.

b) Modular and extensible design: The system architecture should be modular and extensible, allowing for the integration of new sensors, functionality, or communication protocols as they become available. This modular approach ensures scalability and adaptability to evolving healthcare requirements and technological advancements.

c) Power efficiency and battery management: For practical and sustainable use, the system should prioritize power efficiency and effective battery management. Selecting low-power components, optimizing power consumption algorithms, and incorporating rechargeable battery solutions will contribute to the system's longevity and environmental sustainability.

d) Simplicity of maintenance and repair: Considerations should be given to the simplicity of maintenance and repair of the system components. Open-source hardware and software facilitate access to documentation, community support, and readily available replacement parts, reducing the need for frequent replacements and fostering a circular economy approach.

e) User-friendly interfaces and data visualization: Effective user interfaces and data visualization tools are crucial for the system's adoption and usability. Evaluating user experience design principles and integrating intuitive graphical interfaces will enhance user engagement and promote long-term adherence to the health monitoring system.

f) Interoperability and data standardization: To ensure seamless integration with existing healthcare systems and facilitate data sharing, the autonomous health monitoring system should adhere to industry standards and data interoperability protocols. This will facilitate collaboration, data exchange, and prospective integration with electronic health records (EHRs) or telemedicine platforms.

g) Scalability and cloud integration: Evaluating cloud-based solutions for data storage, processing, and analysis can provide scalability and remote access capabilities. Cloud integration can facilitate seamless data synchronization, remote monitoring, and the potential for machine learning and artificial intelligence applications in health data analysis.

By meticulously selecting and evaluating tools and technologies based on these factors, the development of an automatic health monitoring system using Arduino can accomplish sustainable and long-term viability, adaptability, and impact in the healthcare domain.

3.3 Project Work Plan

Proper planning is necessary to ensure the project is completed by the deadline and has the intended operational capacity. The project would require two semesters, or thirty weeks, to complete. Additionally, submitting this project is crucial to ensuring that Project Sarjana Muda 1 and Project Sarjana Muda 2 have the desired results and reducing the likelihood of failure.

The report's progress is depicted in the Gantt chart, which is displayed in table 3.1. The Gantt chart in Table 3.1 illustrates the project's progress starting with the project title selection and ending with the process used to finish the PSM report. The Gantt chart also shows the utilization of software and hardware components.

Table 3.1: Gantt Chart for PSM 1 and PSM 2

No.	Activities	PSM 1														Activities	PSM 2													
		Weeks															Weeks													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14		1	2	3	4	5	6	7	8	9	10	11	12	13	14
1.	PSM title and supervisor registration	■							Midterm Break							Develop the hardware	■	■						Midterm Break						
2.	Problem Statement		■													Hardware testing		■												
3.	Project Objectives		■													IoT Application (Blynks)			■											
4.	Preparation Chapter 1			■												Perform the experiment				■										
5.	Submission of Chapter 1				■											Record the data				■										
6.	Literature Review				■	■	■	■		■	■	■	■			Analyze the result					■									
7.	Software and Hardware specification					■										Prepation of Chapter 4					■	■								
8.	Preparation Chapter 2						■	■								Submission of Chapter 4							■							
9.	Submission of Chapter 2									■						Final Hardware Preparation									■					
10.	Preparation of Chapter 3										■					Preparation of Chapter 5									■	■				
11.	Perform Circuit Design										■					Submission of Chapter 5										■				
12.	Submission of Chapter 3											■				Poster Preparation										■				
13.	Write a coding												■			Poster Submission											■			
14.	Submission of Chapter 4												■			Preparing for Project Final report												■		
15.	PSM 1 Final report submission + turnitin													■		PSM 2 Final report submission + turnitin													■	
16.	PSM 1 Presentation														■	PSM 2 Presentation														■

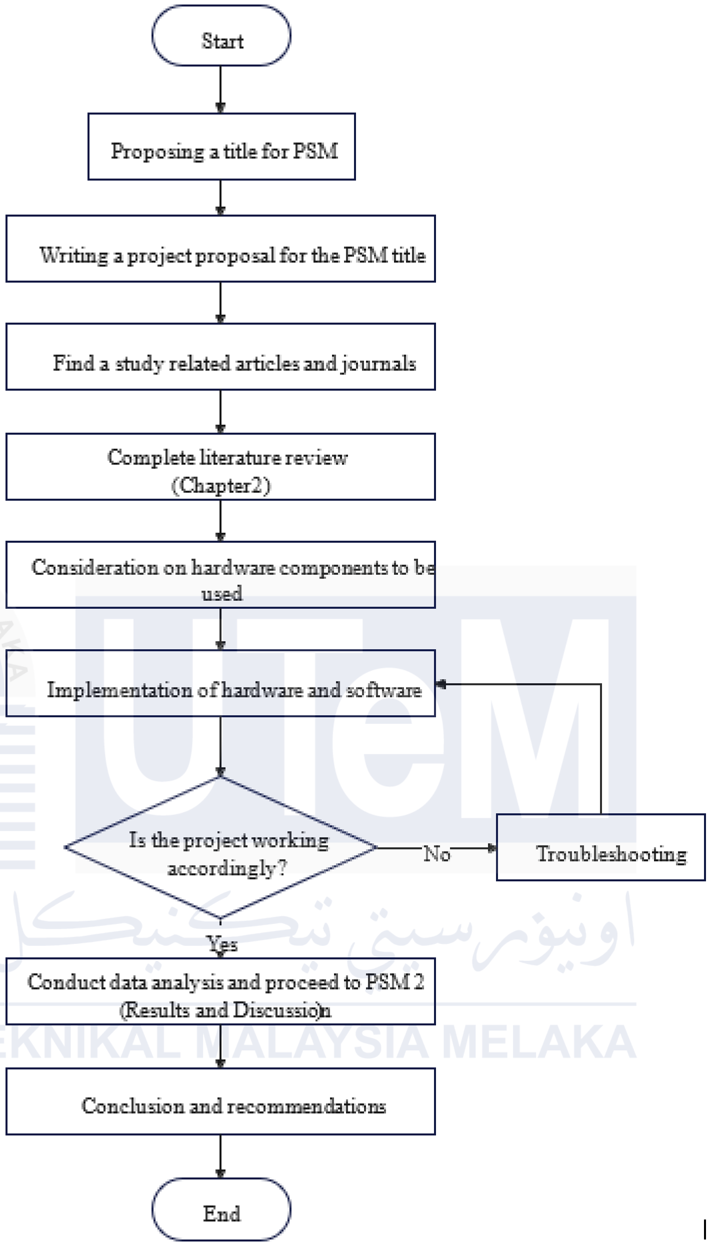

Figure 3.1: PSM Flow Flowchart

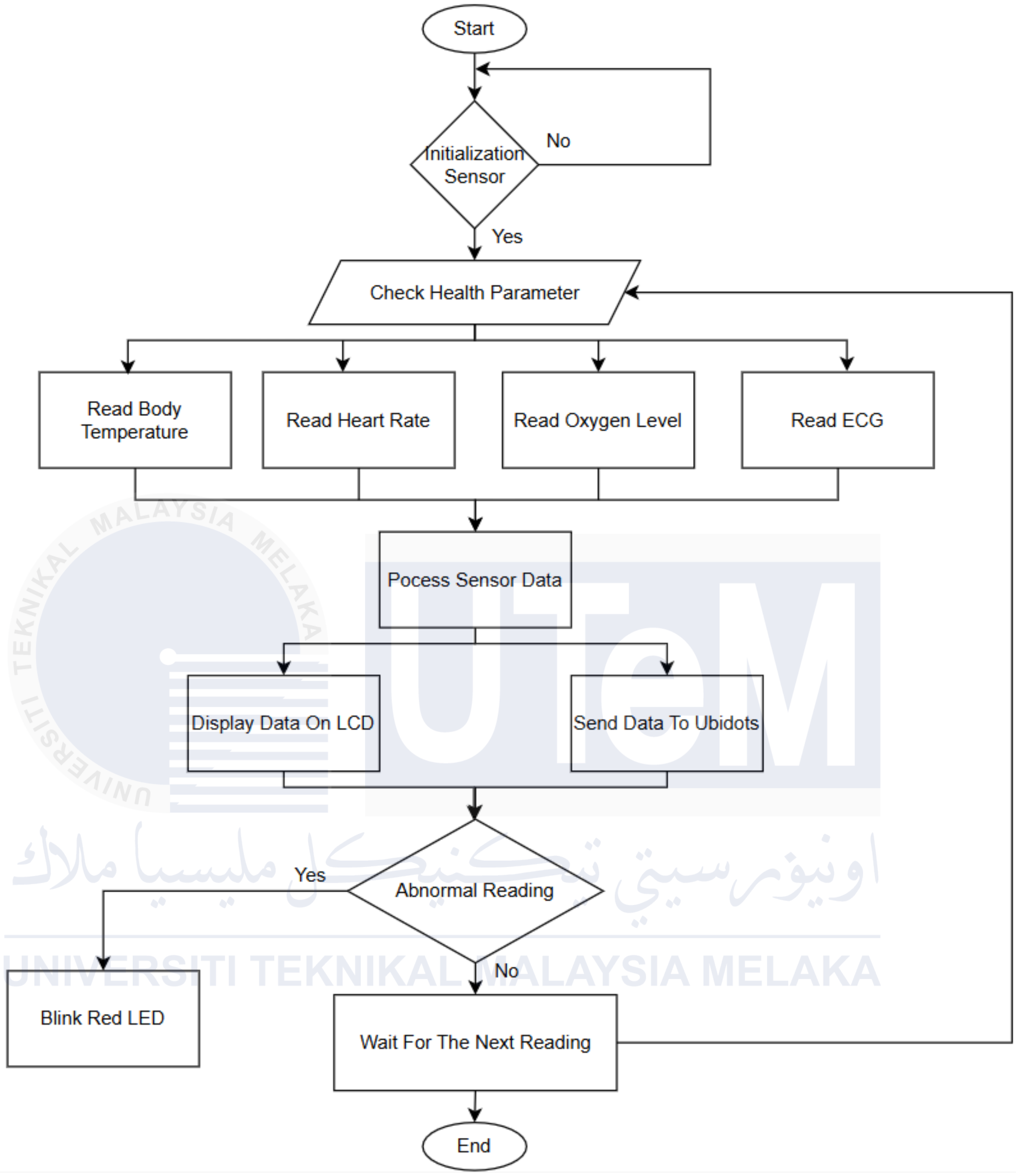

Figure 3.2 Flowchart of the project

3.4 Project Block Diagram

The project block diagram for "Automatic Health Monitoring Using Arduino" serves as a visual representation of the system's architecture and functionalities. This innovative project integrates Arduino, a widely-used microcontroller platform, with various sensors and modules to monitor and track an individual's health parameters in real-time.

The block diagram outlines the key components and their interactions within the system. At the core is the Arduino microcontroller, functioning as the central processing unit. Connected to it are a series of sensors responsible for collecting vital health data. These sensors may include but are not limited to:

- ECG Sensor: Measures the user's heart rate continuously, providing insights into cardiovascular health.
- LM 35 Sensor: Monitors body temperature variations, aiding in the early detection of fever or abnormal temperature fluctuations.
- MAX 30102 Sensor: This device measures pulse rate and blood oxygen saturation levels, two vital signs of respiratory health.

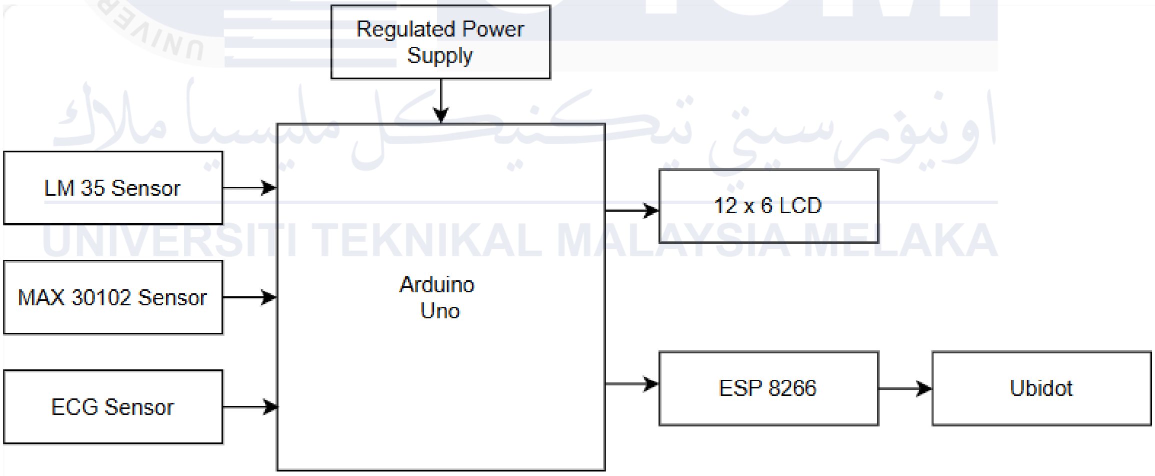

Figure 3.3 Block diagram of automatic health monitoring system using arduino

3.5Hardware Specification

This part shows that there were many hardware used in order to achieve the desired output of the project. Each hardware element is carefully chosen to ensure accuracy, reliability, and compatibility with the Arduino platform, facilitating seamless integration and operation.

3.5.1 Arduino Uno

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button [18]. The Arduino Uno is versatile and may be used in many applications such as robotics, home automation, interactive art displays, and more. The software's user-friendly interface, wide-ranging community assistance, and budget-friendly cost make it a superb option for both novices and proficient users.

Figure 3.4 Arduino uno

Table 3.2 Specification of arduino uno

Specification	Technical Condition
Input / Output Pins	14 Pins
Analog Inputs	6 Inputs
Quartz Crystal	16MHz

Input Voltage	5V
Reset Button	-

3.5.2 MAX30102 – Pulse Oximeter and Heart Rate Sensor

The MAX30102 is an integrated pulse oximetry and heartrate monitor sensor solution. It combines two LEDs, a photodetector, optimized optics, and low-noise analog signal processing to detect pulse oximetry and heart-rate signals [19]. The MAX30102 is extensively utilized in a range of projects, including wearable health monitors, fitness trackers, and medical gadgets. The compact dimensions, minimal energy consumption, and user-friendly nature of this device render it a favoured option among both enthusiasts and experts engaged in health-oriented endeavours.

Figure 3.5 MAX30102 Pulse Oximeter and Heart Rate Sensor

Table 3.3 Specification of MAX30102 Pulse Oximeter and Heart Rate Sensor

Specification	Details
Physiological Parameters	Pulse Oximetry (SpO2) range: 0-100% Heart Rate range: 30 bpm to 250 bpm (beats per minute).
Interface	12C (SDA/SCL)
Operating Voltage	1.8V to 3.3V (DC).

3.5.3 LM35 - Temperature Sensor

The LM35 is a very accurate temperature sensor that was created by Texas Instruments. Its precision, linearity, and ease of use make it frequently utilized for temperature measuring in numerous applications. The LM35 is a very accurate temperature sensor that was created by Texas Instruments. Its precision, linearity, and ease of use make it frequently utilized for temperature measuring in numerous applications.

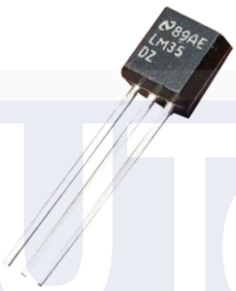

Figure 3.6 LM35 Sensor

Table 3.4 Specifications of LM35 sensor

Specification	Details
Function	Measures body temperature (36.5-37.5°C normal)
Interface	Analog output (10mV/°C)
Operating Voltage	4-30V

3.5.4 ESP8266 - Wi-Fi Module

The ESP8266 is an inexpensive microcontroller with built-in Wi-Fi capabilities that was created by Espressif Systems. It has become widely popular among both enthusiasts and professionals because to its adaptability, small size, and impressive capabilities. The ESP8266 itself is a self-contained WiFi networking solution offering as a bridge from existing micro controller to WiFi and is also capable of running self-contained applications [20].

Figure 3.7 ESP 8266

Table 3.5 Specifications of ESP8266

Specification	Details
Function	Transmits health data to cloud/app
Interface	UART (RX/TX)
Operating Voltage	3.3V
Wi-Fi	802.11 b/g/n

3.5.5 16x2 LCD Display

A 16x2 LCD (Liquid Crystal Display) is a frequently used electronic display module that can show 16 characters per line on 2 lines. Each character is displayed using a 5x8 pixel grid, which allows for displaying letters, integers, and some basic custom characters. Most 16x2 LCDs come with an optional backlight (typically blue or green) that increases visibility in low-light environments.

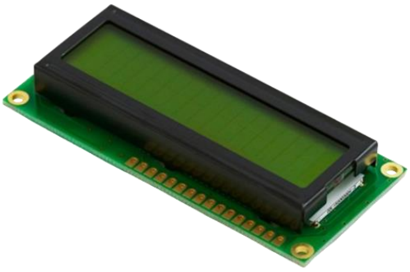

Figure 3.8 16x2 LCD Display

Table 3.6 Specifications of 16x2 LCD Display

Specification	Details
Function	Shows readings (e.g., "SpO2: 98% HR: 72")
Interface	4-bit parallel (4 data, 2 control pins)
Operating Voltage	5V

3.5.6 Resistors

Resistor is electrical or electronic components which resist the flow of current across the resistor device[21]. Resistors are required in practically all electronic circuits for numerous functions, such as current limitation, voltage division, and signal conditioning. They are available in many kinds, values, tolerances, and power ratings, and are utilized in a broad range of applications, from basic LED circuits to complicated filters and oscillators.

Figure 3.9 Resistor of 220 Ω

3.5.7 LED

LED is a semiconductor diode that emits incoherent narrow-spectrum of light when electrically biased in the forward direction of the p-n junction, as in the common LED circuit[22]. LEDs are noted for their high efficiency, extended lifespan, low power consumption, and tiny size. They are available in many colors and forms, and are utilized in a broad range of applications, including lighting, automotive, displays, signs, and electronics.

Figure 3.10 Led

Table 3.7 Specifications of LED

Specification	Details
Function	Status indicator (e.g., green=normal)
Type	5mm, 20mA max

3.5.8 Donut Board

A donut board (perforated prototyping board) serves as a platform for securely soldering electronic components like sensors, microcontrollers, and resistors, enabling the creation of custom circuits[23]. It facilitates durable connections essential for reliable operation, supports scalable design for integrating additional components, and ensures organized circuit assembly. Typically made of fiberglass-reinforced epoxy or phenolic resin, it features a grid of copper-plated holes spaced at 2.54 mm (0.1 inch) intervals, allowing flexible circuit design. Its robust nature makes it ideal for prototyping and testing in health monitoring systems.

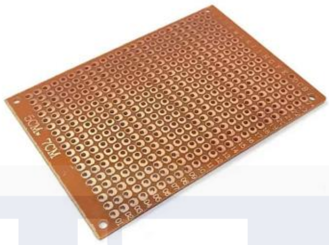

Figure 3.11 Donut Board

Table 3.8 Specification of Breadboard

Specification	Details
Material	Fiberglass-reinforced epoxy (FR4) or phenolic resin.
Board Size	Varies; common sizes include 5 cm × 7 cm, 10 cm × 15 cm, etc.
Connectivity	Holes may be isolated or have copper pads, enabling flexibility in circuit design.

3.5.9 Jumper Wires

Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering[24]. A jumper wire is a thin, flexible wire that is used on a breadboard or other prototyping platform to create temporary electrical connections between components. They are vital for electronics prototyping, enabling for rapid and easy adjustments to circuits. Jumper wires come in many lengths, colors, and varieties, including female-to-female (F-F), male-to-male (M-M), and male-to-female (M-F). They enable flexibility, reusability, and structure in electronic projects. When utilizing jumper wires, it is crucial to pick the suitable length and gauge for the individual application to guarantee maximum performance and minimize signal deterioration or component stress.

Figure 3.12 Jumper wires

Table 3.9 Specifications of jumper wires

Specification	Detail
Function	Make connections
Type	Male-to-male, male-to-female

3.5.10 3.7V Lithium-Ion Battery

A 3.7V lithium-ion battery is widely used in electronic projects, including automatic health monitoring systems, due to its high energy density, lightweight, and rechargeable nature. It provides consistent and reliable power to components like sensors, microcontrollers, communication modules, and display units, ensuring stable operation with its nominal 3.7V output. Its rechargeable capability makes it cost-effective and environmentally friendly, while its compact and lightweight design ensures portability for wearable or portable systems. Additionally, it supports high current discharge, enabling it to power components requiring momentary surges, such as Wi-Fi or Bluetooth modules.

Figure 3.13 3.7V lithium battery

Table 3.10 Specifications of 3.7V lithium battery

Specification	Details
Type	3.7V Lithium-Ion Battery
Nominal Voltage	3.7V
Capacity	Typically ranges from 500 mAh to 5000 mAh or higher, depending on the model.
Suitability	Suitable for short-term use

3.5.11 ECG Sensor

An ECG sensor measures the electrical activity of the heart, playing a vital role in monitoring heart health. It detects and records electrical signals from the heart via electrodes, enabling functions such as heart rate monitoring, real-time analysis, and detection of arrhythmias or abnormalities like bradycardia and tachycardia. It provides real-time data for continuous health monitoring and can be integrated into wearable devices for remote tracking. Designed for low power consumption, it outputs analog or digital signals compatible with microcontrollers, making it ideal for use in portable and battery-operated health monitoring systems.

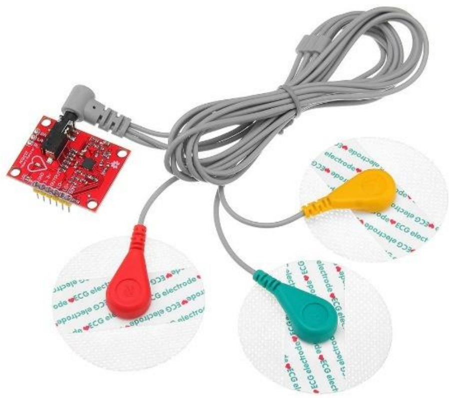

Figure 3.14 ECG Sensor

3.6 Software Specification

This part shows that there were many sofwaree used in order to achieve the desired output of the project. Each hardware element is carefully chosen to ensure accuracy, reliability, and compatibility with the Arduino platform, facilitating seamless integration and operation.

3.6.1 Arduino IDE

The Arduino Software (IDE) is also known as the Arduino Integrated Development Environment. It includes a text editor for writing code, a message box, a text console, a toolbar with buttons for frequently used tasks, and other menus. Arduino IDE is an open source software that is mainly used for writing and compiling the code into the Arduino Module[25] . In order to upload and interact with programs, it establishes a connection with the Arduino hardware.

Figure 3.15 Arduino IDE

3.6.2 Solidworks

The SOLIDWORKS® CAD software is a mechanical design automation application that lets designers quickly sketch out ideas, experiment with features and dimensions, and produce models and detailed drawings[26].

Figure 3.16 Solidworks

3.6.3 Ubidots

Ubidots is an Internet of Things (IoT) cloud platform designed for collecting, visualizing, and analyzing real-time sensor data. Enable data-driven decisions by leveraging Ubidots' codeless visualizations, smart alerts, scheduled reports, data analytics, and more [27]. It provides tools for managing IoT devices, sending data to the cloud, and creating custom dashboards to visualize key metrics. Ubidots is widely used in IoT projects because it simplifies the process of connecting physical devices to the internet, enabling remote monitoring and control of sensors and actuators.

Key features of Ubidots include:

- Data Ingestion: Collects data from a variety of IoT protocols such as HTTP, MQTT, and TCP/UDP.
- Real-Time Dashboards: Customizable dashboards to visualize data with widgets like graphs, indicators, and maps.
- Event Management: Triggers for events like alerts, webhooks, or email notifications based on specified conditions.
- API Integration: RESTful and MQTT APIs for seamless integration with other services.

Figure 3.17 Ubidots

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

This chapter present the results from the health monitoring system designed using Arduino components such as the MAX30102 pulse oximeter, ECG sensor, temperature sensor, and an LCD display. The system provides real-time measurements of heart rate (HR), blood oxygen saturation (SpO2), body temperature, and ECG, while also performing warnings and alerts based on predefined thresholds for each of these parameters.

4.2 System Overview

The health monitoring system continuously gathers data from various sensors to monitor the user's physiological state. The system is composed of several components:

a) MAX30105 Sensor: Measures heart rate (HR) and SpO2 levels using infrared and red LEDs.

b) ECG Sensor: Measures the electrical activity of the heart.

c) Temperature Sensor: Monitors the body temperature using an analog sensor.

d) LCD Display: Displays real-time readings of heart rate, SpO2, temperature, and ECG values.

e) Buzzer and Warning LED: Provide alerts for abnormal readings based on threshold values.

4.3 Experimental Setup

The system was tested under different conditions to observe how it responds to varying physiological parameters. Each component of the system was evaluated for its performance in terms of data accuracy, reliability, and real-time response. The main sensor parameters evaluated include:

a) Heart Rate (HR): The heart rate was measured using the MAX30105 sensor. The system was designed to detect and calculate HR based on the infrared signal received by the sensor, with acceptable values between 40-180 beats per minute (BPM).
b) SpO2: The blood oxygen saturation was also measured by the MAX30105 sensor, using a calculation based on the ratio of red and infrared light absorption. SpO2 values typically range from 70% to 100%.
c) Body Temperature: A temperature sensor placed on the body was used to monitor the user's body temperature, with normal readings ranging from 20°C to 38°C.
d) ECG: The system reads the electrical signals of the heart from the ECG sensor and reports the signal strength in analog values.

4.4 Data Collection and Analysis

4.4.1 Heart Rate

The heart rate was calculated by measuring the time between successive heartbeats detected by the MAX30105 sensor. The system displayed the heart rate on the LCD and used a warning mechanism if the heart rate exceeded predefined thresholds (i.e., HR < 60 BPM or HR > 100 BPM). The heart rate detection algorithm worked effectively for normal ranges. However, if the HR was too low or too high, the system triggered a warning with both visual (LED blink) and auditory (buzzer) feedback.

- Example: If the heart rate was detected as 55 BPM, the system displayed "HR: 55" with an exclamation mark on the LCD, and the warning LED blinked. This response was accurate and in line with expected thresholds.

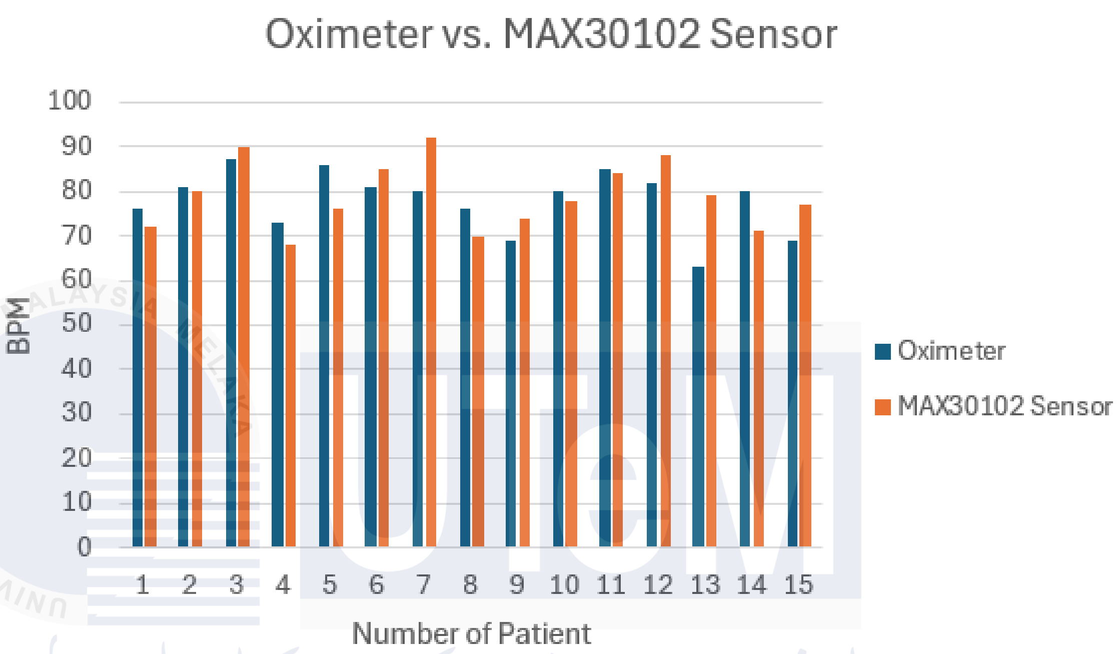

Figure 4.1 Data analysis for heart rate

Table 4.1 Data collection of three readings for 15 patients for heart rate.

Patient	First reading heart rate (BPM)	Second reading heart rate (BPM)	Third reading heart rate (BPM)
1	72	75	70
2	80	78	82
3	90	85	88
4	68	70	72
5	76	78	74
6	85	83	87
7	92	89	91
8	70	72	71
9	74	76	77
10	78	75	80
11	84	82	85
12	88	86	90
13	79	81	78

14	71	73	72
15	77	80	75

4.4.2 Oxygen Saturation (SpO2)

SpO2 was calculated using the ratio of red and infrared light absorption and displayed on the LCD. The system was able to compute SpO2 levels accurately when the finger was placed correctly on the MAX30105 sensor, with the system updating SpO2 values every second. The system provided an alert if SpO2 dropped below 90%.

- Example: If SpO2 dropped to 85%, the display showed “SpO2: 85%!” with a visual warning on the LCD and the warning LED blinking.

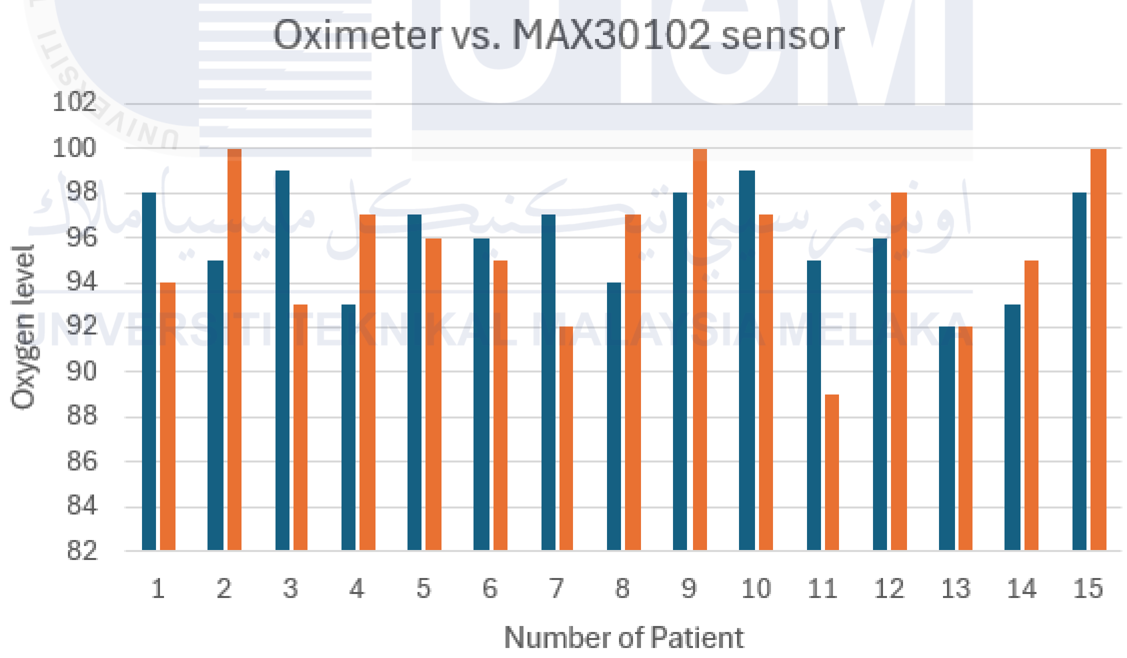

Figure 4.2 Data analysis for oxygen level

Table 4.2 Data collection of three readings for 15 patients for oxygen level.

Patient	First reading SpO2	Second reading SpO2	Third reading SpO2
1	98	97	96
2	95	96	94
3	99	98	97
4	93	92	94
5	97	98	95
6	96	95	97
7	97	96	98
8	94	95	93
9	98	97	96
10	99	98	97
11	95	94	96
12	96	97	98
13	92	94	93
14	93	94	91
15	98	97	96

4.4.3 Body Temperature

The temperature sensor gathered analog data, which was processed to determine the body temperature. Using a simple moving average of multiple readings, the system was able to provide stable temperature values. The temperature warning mechanism was triggered if the body temperature exceeded 38°C (fever warning) or dropped below 20°C (hypothermia warning).

- Example: If the temperature was recorded at 39°C, the display showed “T: 39°C!” with an exclamation mark, indicating a fever condition.

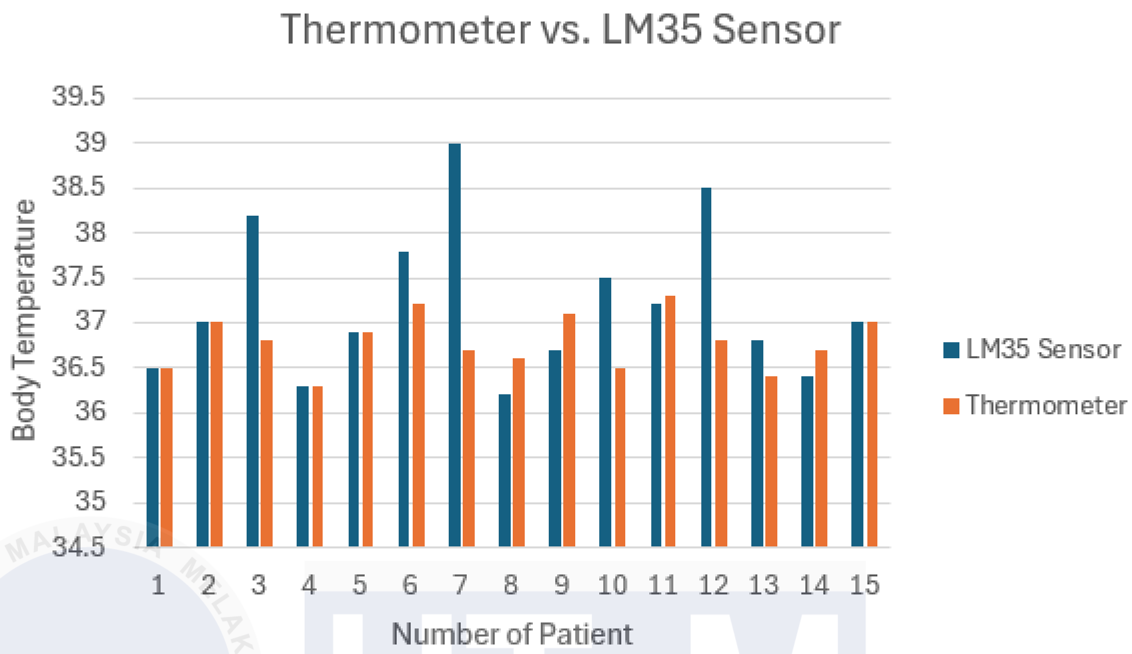

Figure 4.3 Data analysis for body temperature

Table 4.3 Data collection of three readings for 15 patients for body temperature.

Patient	First reading body temperature	Second reading body temperature	Third reading body temperature
1	36.5	36.7	36.4
2	37.0	36.8	37.1
3	36.8	37.0	36.9
4	36.3	36.5	36.6
5	36.9	37.0	36.7
6	37.2	37.3	37.1
7	36.7	36.9	37.0
8	36.6	36.8	36.5
9	37.1	37.2	37.0
10	36.5	36.7	36.6
11	37.3	37.4	37.2
12	36.8	37.0	36.9
13	36.4	36.6	36.5
14	36.7	36.8	36.5
15	37.0	37.1	36.9

4.4.4 Electrocardiogram (ECG)

The ECG sensor was monitored for signal strength to detect heart activity. The system displayed the ECG value on the LCD and triggered a warning if the ECG reading went beyond acceptable thresholds (ECG < 100 or ECG > 900). The system also checked the connection status of the ECG leads, and in case of a disconnected lead, the ECG value was displayed as zero, while a warning was shown.

- Example: If the ECG value was zero due to a loose connection, the system displayed "E: 0" and triggered a warning LED and buzzer alert.

4.4.5 ECG Mismatch Between Arduino and Ubidots

4.4.5.1 Understanding the Issue

a) Real-Time Data Collection and Display:

The ECG data is collected and displayed in real-time on the arduino at a higher frequency than the data transmission to Ubidots. Each time the system captures a new ECG reading, it updates the arduino display immediately, often capturing multiple data points per second.

b) Data Transmission to Ubidots:

The system is designed to send data to Ubidots once every second (due to the limitation of the communication frequency). This means that only one ECG reading is transmitted per second. As a result, when the system continuously collects ECG data at a higher rate, only the latest value is sent to Ubidots, while earlier readings in that second are missed. This leads to a mismatch between the arduino display and the data that appears on the Ubidots graph.

4.3.5.2 Causes of Mismatch

a) Transmission Rate Limitation:

Cause: Ubidots allows only one data point per second to be transmitted for each sensor parameter. This limitation forces the system to discard any additional readings that occur within the same second.

Effect: As a result, multiple ECG data points are ignored within each second, and only the most recent value is sent to Ubidots. This causes the graph on Ubidots to appear less granular and might cause visual discrepancies when compared to the real-time ECG readings displayed on the arduino.

b) Data Update Frequency:

Cause: The LCD is updated frequently, and the ECG value is updated every few milliseconds depending on the sensor's sampling rate. The system might capture ECG variations within a second, leading to a finer resolution on the arduino.

Effect: These variations are not reflected in the data sent to Ubidots because only one value is recorded every second. The graph on Ubidots may not show the ECG fluctuations seen on the arduino, making it appear smoothed or disconnected.

c) Data Discrepancy Between Display and Graph:

Cause: The graph on Ubidots is designed to plot ECG data continuously over time. However, due to the transmission rate limitation, this graph might display intervals of missing data, creating gaps or flat lines at certain points.

Effect: These gaps or discrepancies between the arduino and Ubidots may mislead users into thinking the ECG signal is not updating correctly in real-time, even though it is accurately being captured and displayed on the arduino.

4.4.5.3 Impact of Missing Data

a) Graph Discontinuities: Since Ubidots only receives one ECG value per second, multiple ECG fluctuations are lost within each second, leading to a jagged or step-like appearance on the graph, rather than a continuous curve that accurately represents the heart's electrical activity.

b) Lag Between Real-Time Data and Ubidots: The delay caused by sending only one data point per second results in a noticeable lag between what is displayed on the arduino and what appears on the Ubidots graph. Users may see the ECG value change in real-time on the arduino, but the graph on Ubidots will be updated with a delay, making it look like the system is not providing real-time feedback.

4.5 Threshold and Warning Mechanisms

The system includes predefined thresholds for each physiological parameter, and any abnormal readings trigger warnings to alert the user. The thresholds are as follows:

- Temperature: Below 20°C or above 38°C.
- Heart Rate: Below 60 BPM or above 100 BPM.
- SpO2: Below 90%.
- ECG: Below 100 or above 900 (indicative of sensor disconnection or extreme signal interference).

Whenever any of these thresholds were crossed, the system provided a visual and auditory warning using an LED that blinked and a buzzer that emitted a short beep.

4.6 Data Transmission to NodeMCU

The system also sends collected data (heart rate, SpO2, temperature, and ECG) to a NodeMCU module via serial communication. Data was transmitted in a compact JSON format, ensuring efficient communication and reducing latency. This data could be used for further processing, such as monitoring the user remotely or integrating the data into a health management system

4.7 Summary

Chapter 4 presents the results of a health monitoring system using Arduino components like the MAX30102 pulse oximeter, ECG sensor, temperature sensor, and an LCD display, which measures heart rate (HR), blood oxygen saturation (SpO2), body temperature, and ECG in real-time while providing alerts for abnormal readings. The system effectively monitors HR, SpO2, body temperature, and ECG, with predefined thresholds triggering visual and auditory warnings when values fall outside acceptable ranges. However, a mismatch between the Arduino display and the data sent to Ubidots is observed due to Ubidots limitation of transmitting only one data point per second, leading to data loss and gaps in the Ubidots graph. Despite these discrepancies, the system accurately captures and displays data in real-time on the Arduino, while the NodeMCU transmits compact JSON data for remote monitoring. The system demonstrates reliable performance, although the graph on Ubidots appears less granular and may mislead users about the real-time updates due to the transmission rate limitation.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The primary objectives that were established at the inception of the project have been satisfactorily realized through the development of an autonomous health monitoring system utilizing Arduino. The Arduino board was able to seamlessly integrate a variety of vital sign sensors, such as temperature, pulse, oxygen level, and ECG sensors, due to the meticulous design and implementation of the system architecture.

Rigorous testing and evaluation were implemented throughout the development process to guarantee the system's dependability and accuracy. The system's user-friendliness was also a critical factor, and efforts were made to develop an intuitive and accessible interface for the purpose of monitoring and interpreting the collected health data.

The automatic health monitoring system has the potential to transform the way individuals monitor their health, enabling them to take proactive measures to maintain their well-being. The system can facilitate the early detection of prospective health issues and the provision of timely intervention or medical attention when necessary by providing real-time data on vital signs.

Furthermore, the system's adaptability to changing healthcare requirements is guaranteed by the modular design, which enables the future integration of additional sensors or functionalities. The Arduino-based autonomous health monitoring system's successful development is a testament to the potential of open-source hardware and software to transform healthcare technology.

In the future, additional research and development initiatives could concentrate on improving the system's accuracy, broadening its spectrum of measurable vital signs, and investigating cloud-based data storage and analysis capabilities. Furthermore, the system's pervasive adoption and practical application in a variety of healthcare settings will be guaranteed by field testing and user feedback, which will be invaluable in refining the system.

5.2 Recommendations

As the healthcare business advances, embracing technology breakthroughs like the autonomous health monitoring system using Arduino and ESP8266 is vital for enhancing patient care and results. While first project stages showed promising results, this recommendations section addresses upgrades to this project. For instance:

a) Improve User Interface and Data Visualization: The system's user interface and data visualization capabilities might be expanded to increase usability and accessibility. This may require designing a user-friendly mobile or web application that presents real-time data and warnings in a straightforward and readily comprehensible way.

b) Enhance Security and Data Privacy: Implement strong security measures to secure patient data and maintain compliance with data privacy legislation. This may require encrypting data communications, creating secure authentication procedures, and complying to industry standards such as HIPAA (Health Insurance Portability and Accountability Act) or GDPR (General Data Protection Regulation).

c) Develop a Mobile Application: Create a user-friendly mobile application that acts as an interface for patients, caregivers, and healthcare professionals. The app might show real-time health data, issue alarms, and offer access to past information and reports.

d) Integrate with Existing Healthcare Systems: Investigate potential for integration with existing healthcare systems, such as electronic medical records (EMR) or hospital information systems. This will enable seamless data exchange and collaboration across healthcare providers, enhancing patient care and simplifying workflows.

REFERENCES

[1] World Health Organization, "Noncommunicable diseases," *World Health Organization*, Jan. 21, 2025. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/noncommunicable-diseases. [Accessed: Jan. 21, 2025].

[2] R. Ohannessian, T. A. Duong, and A. Odone, "Global Telemedicine Implementation and Integration Within Health Systems to Fight the COVID-19 Pandemic: A Call to Action," *JMIR Public Health and Surveillance*, vol. 6, no. 2, p. e18810, Apr. 2020, doi: https://doi.org/10.2196/18810.

[3] "Majumder, S., Aghayi, E., Noferesti, M., et al. (2017) Smart Homes for Elderly Healthcare-Recent Advances and Research Challenges. Sensors, 17, 2496. - References - Scientific Research Publishing," *www.scirp.org*. https://www.scirp.org/reference/referencespapers?referenceid=2506077

[4] S. B. Omer, P. Malani, and C. del Rio, "The COVID-19 Pandemic in the US: A Clinical Update," *JAMA*, vol. 323, no. 18, Apr. 2020, doi: https://doi.org/10.1001/jama.2020.5788.

[5] L. F. Indust, "IoT Patient Health Monitoring with ESP8266 & Arduino," pp. 1–30, 2024, [Online]. Available: https://how2electronics.com/iot-patient-health-monitoring-with-esp8266-arduino/

[6] K. Zhang and W. L. P. U.- 2020, "Health Monitoring of Human Multiple Physiological Parameters Based on Wireless Remote Medical System," *IEEE Access*, doi: https://doi.org/10.1109/access.2020.2987058.

[7] M. M. Khan, T. M. Alanazi, A. A. Albraikan, and F. A. A. P. U.- 2022, "IoT-Based Health Monitoring System Development and Analysis," *Security and Communication Networks*, doi: https://doi.org/10.1155/2022/9639195.

[8] A. Lopez, Y. A. Jiménez, R. Bareno, B. Balamba, and J. F. G. S. P. U.- 2019, "E-Health System for The Monitoring, Transmission and Storage of The Arterial Pressure of Chronic-Hypertensive Patients", doi: https://doi.org/10.1109/coniiti48476.2019.8960803.

[9] C. Nwibor *et al.*, "Remote Health Monitoring System for the Estimation of Blood Pressure, Heart Rate, and Blood Oxygen Saturation Level," *IEEE Sens J*, doi: https://doi.org/10.1109/jsen.2023.3235977.

[10] M. P. Savaridass, N. Ikram, R. Deepika, and R. Aarnika, "Development of smart health monitoring system using Internet of Things," *Mater Today Proc*, vol. 45, pp. 986–989, 2021, doi: 10.1016/j.matpr.2020.03.046.

[11] Preeti and C. R. P. U.- 2022, "IoT and Cloud Based health monitoring system Using Machine learning", doi: https://doi.org/10.1109/ic3i56241.2022.10072946.

[12] Salini Pradhan, Ahmad Anwar Zainuddin, Rohilah Sahak, and M. F. A. Mohamad Yunos, "Investigation into Smart Healthcare Monitoring System in an IoT Environment,"

Malaysian Journal of Science and Advanced Technology, vol. 2, no. 2, pp. 73–77, 2022, doi: 10.56532/mjsat.v2i2.53.

[13] K. Yadav, A. H. Alharbi, A. Jain, and R. A. R. P. U.- 2022, “An IoT Based Secure Patient Health Monitoring System,” *Computers, materials & continua*, doi: https://doi.org/10.32604/cmc.2022.020614.

[14] M. Z. Rahman, M. A. Akbar, V. Leiva, A. Tahir, M. T. Riaz, and C. Martin-Barreiro, “An intelligent health monitoring and diagnosis system based on the internet of things and fuzzy logic for cardiac arrhythmia COVID-19 patients,” *Comput Biol Med*, vol. 154, no. September 2022, p. 106583, 2023, doi: 10.1016/j.compbiomed.2023.106583.

[15] V. R. Parihar, “Automatic wireless health monitoring system,” vol. 1, no. 1, 2018.

[16] M. P. Savaridass, N. Ikram, R. Deepika, and R. A. P. U.- 2021, “Development of smart health monitoring system using Internet of Things,” *Mater Today Proc*, doi: https://doi.org/10.1016/j.matpr.2020.03.046.

[17] V. Akhila, Y. Vasavi, K. Nissie, and P. V. Rao, “An IoT based Patient Health Monitoring System using Arduino Uno,” *International Journal of Research in Information Technology*, vol. 1, no. 1, pp. 2001–5569, 2017, [Online]. Available: www.ijrit.net

[18] T. N. Avilés, “Arduino”.

[19] Maxim Integrated, “Pulse Oximeter and Heart-Rate Sensor IC for Wearable Health,” *Lecture Notes in Energy*, vol. 38, pp. 1–29, 2014, [Online]. Available: www.maximintegrated.com

[20] Started Getting Guide User, “Handson Technology User Manual V1.3 ESP8266 NodeMCU WiFi Development Board Getting Started User Guide,” pp. 1–24, 2018, [Online]. Available: www.handsontec.com

[21] W. District and N. T. City, “What is a resistor?,” *Radio and Electronics Cookbook*, pp. 24–26, 2001, doi: 10.1016/b978-0-08-051644-8.50010-1.

[22] S. C. Singh, “Basics of light emitting diodes, characterizations and applications,” *Handbook of Light Emitting and Schottky Diode Research*, no. May, pp. 133–168, 2009.

[23] A. K. Singh, P. S. Rana, and M. Sharma, "Design and development of a health monitoring system using low-cost prototyping tools," *2019 IEEE International Conference on Healthcare Innovations and Point-of-Care Technologies (HI-POCT)*, Bethesda, MD, USA, 2019, pp. 1-6, doi: 10.1109/HI-POCT45235.2019.8962843.

[24] M. Hemmings, “Jumper Wire,” 2018, [Online]. Available: https://blog.sparkfuneducation.com/what-is-jumper-wire#:~:text=But what are they%3F,Fairly simple.

[25] M. Fezari and A. A. D. Al Zaytoona, “Integrated Development Environment ‘IDE’ For Arduino Integrated Development Environment ‘IDE’ For Arduino Introduction to Arduino

IDE," *ResearchGate*, no. October, 2018, [Online]. Available: https://www.researchgate.net/publication/328615543

[26] "Introducing solidworks /".

[27] Ubidots, "IoT Development Platform," Ubidots, 2025. [Online]. Available: https://ubidots.com. [Accessed: Jan. 21, 2025].

APPENDICES

APPENDIX A

```
#include <ESP8266WiFi.h>
#include <UbidotsESPMQTT.h>
#include <SoftwareSerial.h>
#include <ArduinoJson.h>

// Serial communication with Arduino
SoftwareSerial nodemcu(D5, D6);  // RX, TX

// WiFi and Ubidots credentials
#define WIFI_SSID "eSpace EGW_329B"
#define WIFI_PASS "Admin@123"
#define TOKEN "BBUS-juebyyZrDsAW9fA3r0cyMtPFRg7AOA"   // Replace with your Ubidots TOKEN
#define DEVICE_LABEL "health-monitor"  // Device label to use in Ubidots
```



```
// Constants
#define CONNECTION_TIMEOUT 10000
const int STATUS_LED = LED_BUILTIN;

// Variable labels for Ubidots
#define VAR_HEART_RATE "heart_rate"
#define VAR_SPO2 "spo2"
#define VAR_TEMPERATURE "temperature"
#define VAR_ECG "ecg"
#define VAR_WARNING "warning"

// Initialize Ubidots client
```

```
Ubidots client(TOKEN);

// Connection management
unsigned long lastConnectionAttempt = 0;
unsigned long lastUbidotsUpdate = 0;
const unsigned long UBIDOTS_UPDATE_INTERVAL = 1000;  // Update interval in milliseconds

// Buffer for incoming data
char inputBuffer[150];
int bufferIndex = 0;
bool startFlag = false;

// Data storage
float lastHeartRate = 0;
float lastSpO2 = 0;
float lastTemp = 0;
int lastEcg = 0;
bool lastWarning = false;

void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message received: ");
  for (int i = 0; i < length; i++) {
    Serial.print((char)payload[i]);
  }
  Serial.println();
}

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

```
nodemcu.begin(115200);  // High baud rate for faster transfer

pinMode(STATUS_LED, OUTPUT);
digitalWrite(STATUS_LED, LOW);

// Connect to WiFi
Serial.println("\nConnecting to WiFi");
WiFi.begin(WIFI_SSID, WIFI_PASS);

unsigned long startAttempt = millis();
while (WiFi.status() != WL_CONNECTED && millis() - startAttempt < CONNECTION_TIMEOUT) {
  digitalWrite(STATUS_LED, !digitalRead(STATUS_LED));
  delay(100);
  Serial.print(".");
}

if (WiFi.status() == WL_CONNECTED) {
  Serial.println("\nWiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  digitalWrite(STATUS_LED, HIGH);
} else {
  Serial.println("\nWiFi connection failed!");
  digitalWrite(STATUS_LED, LOW);
}

// Initialize Ubidots connection
client.setDebug(true);  // Enable debug messages
client.begin(callback);
```

```
  if(!client.connected()) {
    client.reconnect();
  }
}

void loop() {
  if(!client.connected()) {
    client.reconnect();
    digitalWrite(STATUS_LED, LOW);
  } else {
    digitalWrite(STATUS_LED, HIGH);
  }

  client.loop();

  // Read data from Arduino
  while (nodemcu.available()) {
    char c = nodemcu.read();

    if (c == '#') {
      startFlag = true;
      bufferIndex = 0;
      continue;
    }

    if (startFlag) {
      if (c == '~') {
        inputBuffer[bufferIndex] = '\0';
        processJsonData(inputBuffer);
```

```
            startFlag = false;
        } else if (bufferIndex < sizeof(inputBuffer) - 1) {
            inputBuffer[bufferIndex++] = c;
        }
    }
}

// Update Ubidots at regular intervals
if (millis() - lastUbidotsUpdate >= UBIDOTS_UPDATE_INTERVAL) {
    updateUbidots();
    lastUbidotsUpdate = millis();
}
}

void processJsonData(const char* jsonString) {
  StaticJsonBuffer<200> jsonBuffer;
  JsonObject& root = jsonBuffer.parseObject(jsonString);

  if (!root.success()) {
    Serial.println("JSON parsing failed!");
    return;
  }

  // Extract and store values
  if (root.containsKey("h")) lastHeartRate = root["h"];
  if (root.containsKey("s")) lastSpO2 = root["s"];
  if (root.containsKey("t")) lastTemp = root["t"];
  if (root.containsKey("e")) lastEcg = root["e"];
  if (root.containsKey("w")) lastWarning = root["w"];
```

```
  // Debug print
  Serial.print("HR: "); Serial.print(lastHeartRate);
  Serial.print(" SpO2: "); Serial.print(lastSpO2);
  Serial.print(" Temp: "); Serial.print(lastTemp);
  Serial.print(" ECG: "); Serial.print(lastEcg);
  Serial.print(" Warning: "); Serial.println(lastWarning);
}

void updateUbidots() {
  if (client.connected()) {
    client.add(VAR_HEART_RATE, lastHeartRate);
    client.add(VAR_SPO2, lastSpO2);
    client.add(VAR_TEMPERATURE, lastTemp);
    client.add(VAR_ECG, lastEcg);
    client.add(VAR_WARNING, lastWarning);

    client.ubidotsPublish(DEVICE_LABEL);
    Serial.println("Data sent to Ubidots");
  }
}

// Error handling function
void handleError(const char* errorMessage) {
  Serial.println(errorMessage);
  digitalWrite(STATUS_LED, LOW);
  delay(1000);
  digitalWrite(STATUS_LED, HIGH);
}

// WiFi connection check
```

```
bool checkWiFiConnection() {
  if (WiFi.status() != WL_CONNECTED) {
    Serial.println("WiFi connection lost!");
    WiFi.reconnect();
    return false;
  }
  return true;
}
```

APPENDIX B

```
#include <Wire.h>
#include <ArduinoJson.h>
#include "MAX30105.h"
#include "heartRate.h"
#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>

// Software serial for NodeMCU communication at higher baud rate
SoftwareSerial nodemcuSerial(2, 3); // RX, TX

// Pin definitions
const int ECG_PIN = A2;
const int LO_PLUS = 5;
const int LO_MINUS = 6;
const int TEMP_PIN = A0;
const int BUZZER_PIN = 12;
const int WARNING_LED_PIN = A1;

// Health Thresholds
const float TEMP_LOW_THRESHOLD = 20.0;    // °C
const float TEMP_HIGH_THRESHOLD = 38.0;   // °C
const int HR_LOW_THRESHOLD = 60;          // BPM
const int HR_HIGH_THRESHOLD = 100;        // BPM
const int SPO2_LOW_THRESHOLD = 90;        // %
const int ECG_LOW_THRESHOLD = 100;        // Analog value
const int ECG_HIGH_THRESHOLD = 900;       // Analog value

// Communication constants
const unsigned long SEND_INTERVAL = 50;  // Increased frequency for faster updates
```

```
const unsigned long DISPLAY_INTERVAL = 100;  // Display update interval

// LED Blinking patterns
const unsigned long LED_BLINK_INTERVAL = 500;
unsigned long lastLedToggle = 0;
bool ledState = false;

// Temperature sampling constants
const int TEMP_SAMPLES = 5;  // Reduced for faster response
const int TEMP_SAMPLE_DELAY = 5;

// Heart Rate Constants
const byte RATE_SIZE = 4;
const int MIN_ACCEPTABLE_HR = 40;
const int MAX_ACCEPTABLE_HR = 180;

// Objects
MAX30105 particleSensor;
LiquidCrystal_I2C lcd(0x27, 16, 2);

// Heart rate variables
byte rates[RATE_SIZE];
byte rateSpot = 0;
long lastBeat = 0;
float beatsPerMinute;
int beatAvg = 0;
long irValue = 0;
const unsigned long FINGER_THRESHOLD = 7000;

// SpO2 variables
```

```
double redDC = 0, irDC = 0;
double spO2 = 0;
bool dcInitialized = false;

// Temperature variables
float tempSamples[TEMP_SAMPLES];
int tempSampleIndex = 0;
unsigned long lastTempReading = 0;
const unsigned long TEMP_READ_INTERVAL = 50;  // Faster temperature readings

// Warning states
bool tempWarning = false;
bool hrWarning = false;
bool spo2Warning = false;
bool ecgWarning = false;

// Timing variables
unsigned long lastSampleTime = 0;
unsigned long lastSendTime = 0;
unsigned long lastDisplayUpdate = 0;
unsigned long lastWarningBeep = 0;

// ECG variables
int ecgValue = 0;
bool leadsConnected = true;

void setup() {
  Serial.begin(115200);
  nodemcuSerial.begin(115200);  // High baud rate for faster transfer
```

```
pinMode(WARNING_LED_PIN, OUTPUT);
digitalWrite(WARNING_LED_PIN, LOW);

lcd.begin();
lcd.backlight();
displayStartupMessage();

pinMode(BUZZER_PIN, OUTPUT);
pinMode(LO_PLUS, INPUT);
pinMode(LO_MINUS, INPUT);
pinMode(ECG_PIN, INPUT);

if (!particleSensor.begin(Wire, I2C_SPEED_FAST)) {
  lcd.clear();
  lcd.print("Sensor Error!");
  Serial.println("MAX30105 not found!");
  while (1);
}

// Configure sensor for optimal performance
particleSensor.setup(60, 4, 2, 400, 411, 4096);
particleSensor.setPulseAmplitudeRed(0x0A);
particleSensor.setPulseAmplitudeIR(0x0A);

// Initialize arrays
memset(tempSamples, 0, sizeof(tempSamples));
memset(rates, 0, sizeof(rates));

delay(1000);
lcd.clear();
```

```
  lcd.print("Ready!");
  delay(1000);
}

void loop() {
  // Read all sensors
  irValue = particleSensor.getIR();
  long redValue = particleSensor.getRed();
  float temperature = getAverageTemperature();
  processECG();

  // Process data
  checkWarnings(temperature);
  processHeartRate();
  if (irValue > FINGER_THRESHOLD) {
    calculateSpO2(redValue, irValue);
  } else {
    resetMeasurements();
  }

  // Update outputs
  updateDisplay(temperature);
  sendDataToNodeMCU(temperature);
  handleWarnings();
  updateWarningLED();

  delay(5);  // Minimal delay for system stability
}

void processHeartRate() {
```

```
  if (irValue > FINGER_THRESHOLD && checkForBeat(irValue)) {
    long delta = millis() - lastBeat;
    lastBeat = millis();

    beatsPerMinute = 60 / (delta / 1000.0);

    if (beatsPerMinute >= MIN_ACCEPTABLE_HR && beatsPerMinute <= MAX_ACCEPTABLE_HR) {
      rates[rateSpot++] = (byte)beatsPerMinute;
      rateSpot %= RATE_SIZE;

      beatAvg = 0;
      byte validRates = 0;
      for (byte x = 0; x < RATE_SIZE; x++) {
        if (rates[x] != 0) {
          beatAvg += rates[x];
          validRates++;
        }
      }

      if (validRates > 0) {
        beatAvg /= validRates;
        if (!anyWarnings()) {
          tone(BUZZER_PIN, 1000, 50);  // Shorter beep
        }
      }
    }
  }
}
```

```
void processECG() {
  leadsConnected = (digitalRead(LO_PLUS) == LOW && digitalRead(LO_MINUS) == LOW);
  if (leadsConnected) {
    ecgValue = analogRead(ECG_PIN);
    Serial.println(ecgValue);
  } else {
    ecgValue = 0;
  }
}

void updateWarningLED() {
  if (anyWarnings()) {
    if (millis() - lastLedToggle >= LED_BLINK_INTERVAL) {
      lastLedToggle = millis();
      ledState = !ledState;
      digitalWrite(WARNING_LED_PIN, ledState);
    }
  } else {
    digitalWrite(WARNING_LED_PIN, LOW);
    ledState = false;
  }
}

void checkWarnings(float temperature) {
  tempWarning = (temperature < TEMP_LOW_THRESHOLD || temperature > TEMP_HIGH_THRESHOLD);
  hrWarning = (beatAvg > 0 && (beatAvg < HR_LOW_THRESHOLD || beatAvg > HR_HIGH_THRESHOLD));
  spo2Warning = (spO2 > 0 && spO2 < SPO2_LOW_THRESHOLD);
```

```
  ecgWarning = leadsConnected && (ecgValue < ECG_LOW_THRESHOLD || ecgValue > ECG_HIGH_THRESHOLD);
}

bool anyWarnings() {
  return tempWarning || hrWarning || spo2Warning || ecgWarning;
}

void handleWarnings() {
  if (anyWarnings() && millis() - lastWarningBeep >= 1000) {
    lastWarningBeep = millis();
    tone(BUZZER_PIN, 2000, 100);  // Shorter warning beep
  }
}

void updateDisplay(float temperature) {
  if (millis() - lastDisplayUpdate >= DISPLAY_INTERVAL) {
    lastDisplayUpdate = millis();

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("HR:");
    if (beatAvg > 0) {
      lcd.print(beatAvg);
      if (hrWarning) lcd.print("!");
    } else {
      lcd.print("---");
    }

    lcd.print(" SpO2:");
```

```
      if (spO2 > 0) {
        lcd.print(spO2, 0);
        lcd.print("%");
        if (spo2Warning) lcd.print("!");
      } else {
        lcd.print("--");
      }

      lcd.setCursor(0, 1);
      lcd.print("T:");
      lcd.print(temperature, 1);
      lcd.print((char)223);
      lcd.print("C");
      if (tempWarning) lcd.print("!");

      if (leadsConnected) {
        lcd.print(" E:");
        lcd.print(ecgValue);
        if (ecgWarning) lcd.print("!");
      }
    }
}

void sendDataToNodeMCU(float temperature) {
  if (millis() - lastSendTime >= SEND_INTERVAL) {
    lastSendTime = millis();

    // Create compact JSON object
    StaticJsonBuffer<128> jsonBuffer;  // Reduced buffer size
    JsonObject& json = jsonBuffer.createObject();
```

```
    // Add essential data only
    json["h"] = beatAvg;  // Shortened key names
    json["s"] = (int)spO2;
    json["t"] = temperature;
    json["e"] = ecgValue;
    json["w"] = anyWarnings() ? 1 : 0;

    // Efficient transmission with markers
    nodemcuSerial.write('#');  // Start marker
    json.printTo(nodemcuSerial);
    nodemcuSerial.write('~');  // End marker
  }
}

float getAverageTemperature() {
  if (millis() - lastTempReading >= TEMP_READ_INTERVAL) {
    lastTempReading = millis();
    float rawTemp = 0;
    for(int i = 0; i < 3; i++) {  // Reduced samples
      rawTemp += analogRead(TEMP_PIN);
      delay(2);
    }
    rawTemp = rawTemp / 3.0;
    float tempC = (rawTemp / 1024.0) * 500.0;
    tempSamples[tempSampleIndex] = tempC;
    tempSampleIndex = (tempSampleIndex + 1) % TEMP_SAMPLES;
  }

  float sum = 0;
```

```
    for(int i = 0; i < TEMP_SAMPLES; i++) {
        sum += tempSamples[i];
    }
    return sum / TEMP_SAMPLES;
}

void calculateSpO2(long redValue, long irValue) {
    if (!dcInitialized) {
        redDC = redValue;
        irDC = irValue;
        dcInitialized = true;
        return;
    }

    double R = ((redValue - redDC) / redDC) / ((irValue - irDC) / irDC);
    spO2 = 110 - 25 * R;
    spO2 = constrain(spO2, 70, 100);
}

void resetMeasurements() {
    beatAvg = 0;
    spO2 = 0;
    memset(rates, 0, sizeof(rates));
    rateSpot = 0;
    dcInitialized = false;
}

void displayStartupMessage() {
    lcd.clear();
    lcd.setCursor(0, 0);
```

```
    lcd.print("Health Monitor");
    lcd.setCursor(0, 1);
    lcd.print("Initializing...");
    delay(1000);
}
```