

Development of a Server-less DAQ System for Remote Monitoring of the Vital Signs of a Patient



B.Sc. ELECTRICAL ENGINEERING

Sohail Anjum

BEEM-F20-008

Hamad Rafiq

BEEM-F20-018

Ali Hammas

BEEM-F20-019

Muhammad Samiullah

BEEM-F20-029

Session: 2020 – 2024

Thesis Supervisor:

Dr. Mustafa Shakir

Thesis Co-Supervisor:

Engineer. Adnan Sindhu

DEPARTMENT OF ELECTRICAL ENGINEERING

SUPERIOR UNIVERSITY

LAHORE, PAKISTAN



Development of a Server-less DAQ System for Remote Monitoring of the Vital Signs of a Patient

Submitted to Superior University,

Lahore in Partial fulfillment of the requirements for the award of a degree of

B.SC ELECTRICAL ENGINEERING

Sohail Anjum

BEEM-F20-008

Hamad Rafiq

BEEM-F20-018

Ali Hammas

BEEM-F20-019

Muhammad Samiullah

BEEM-F20-029

Session: 2020-24

Supervisor:

Dr. Mustafa Shakir

Co-Supervisor:

Engineer. Adnan Sindhu

DECLARATION

This statement report is an arrangement of our creative study task. Moreover, additions of other peoples are elaborate, every determination is made to empower this clearly, with due reference to the literature, and acknowledgment of combined research and discussions. We also assure you that this work is the output of our study, except that it was identified by references and free from plagiarism of other people's work.

Signature:

Sohail Anjum, Hamad Rafiq, Ali Hammas, Muhammad Samiullah

Date:

RESEARCH COMPLETION CERTIFICATE

It is ensured that the research work contained in this paper titled “**Development of a Server-less DAQ System for Remote Monitoring of the Vital Signs of a Patient**” has been investigated and carried out by Sohail Anjum, Hamad Rafiq, Ali Hammas, and Muhammad Samiullah for the degree of Bachelor of Science in Electrical Engineering.

Dr. Mustafa Shakir

Chairman, Electrical Engineering Department

Thesis Committee Member

Thesis Committee Member

Thesis Committee Member

ACKNOWLEDGEMENTS

In the name of Allah, who is most beneficial and most merciful.

We thank Allah Almighty who helps us to obtain our target. We are grateful to our parents, who always motivated us and provided us with the ultimate help, love, and care. We would also like to thank our supervisor Dr. Mustafa Shakir and co-supervisor Engineer. Adnan Sindhu for their support and encourage our motivation to work very hard and intelligently. When addressing the classification problem in this research project, we found them very supportive. Undoubtedly, their constructive remarks on our project require us to think of innovative methods and concepts in the Health Care Sector, Telemedicine, the Internet of Things, Cloud Computing and beyond. We also want to express our appreciation to all instructors who were very helpful and offered valuable assistance, support, and guidance.

Thank you, Sohail Anjum, Hamad Rafiq, Ali Hammas Muhammad Samiullah.

DEDICATION

This thesis is devoted to my supervisor, Dr. Mustafa Shakir, whose guidance, wisdom, and encouragement have been invaluable throughout this journey. Your dedication to my academic and personal growth has been a source of consistent motivation, and I am thankful for your mentorship.

To my family, whose unwavering affection and support have been my anchor through the highs and lows of this endeavor. Your faith in me has been my greatest motivation, and I can't thank you enough for the penances you have made to assist me with arriving at this achievement.

To my friends, who have been my pillars of strength and joy. Your understanding, patience, and encouragement have been essential in keeping me grounded and focused. Thank you for always being there for me.

ABSTRACT

Data Acquisition in patient monitoring systems plays a vital role in improving staff efficiency in hospitals. In this era, the demand for medical support is gradually increasing with a decrease in medical staff. The death rate in ICUs of developing countries is unexpectedly increasing due to a lack of instant response in severe conditions. In recent years, technological advancements have led to new innovative approaches to revolutionize the healthcare systems. IoT-based event-driven and server-less architecture has filled in prominence to meet the issues of various industries. Server-less Architecture has ruled the industrial sector, particularly in the areas of monitoring, automation, and control. Therefore, in this project data acquisition techniques will be employed to acquire the vital signs of a patient from an off-the-shelf patient monitor. The acquired data will be transmitted to the AWS cloud and the cloud has its DB will enable of the communication of patients to subscribers through API from the AWS cloud to the Smart Phone Application. The front end of the system is a web-based application, where the vital signs are communicated to the doctor. This system will access the data of multiple patients to doctors at the same time.

Key index: Intensive Care Unit (ICU); IoT (Internet of Things); Amazon Web Services (AWS); Data Base (DB); Application Programming Interface (API); Critical Care Unit (CCU); Data Acquisition (DAQ); Message Queuing Telemetry Transport (MQTT); Quality of Service (QoS); Raspberry Pi (RPi); Remote Patient Monitoring (RPM).

LIST OF ACRONYMS

API	Application Programming Interface
AWS	Amazon Web Service
CCU	Cardiac Care Unit / Critical Care Unit
CSP	Cloud Service Providers
DAQ	Data Acquisition
DB	Data Base
ICU	Intensive Care Unit
MQTT	Message Queuing Telemetry Transport
QOS	Quality of Service
RDS	Relational Database service
RPi	Raspberry Pi
RPM	Remote Patient Monitoring
SPICE	Super-fast, Parallel, In-memory Calculation Engine

Table of Content

DECLARATION	3
RESEARCH COMPLETION CERTIFICATE	4
ACKNOWLEDGEMENTS	5
DEDICATION	6
ABSTRACT	7
LIST OF ACRONYMS	8
Table of Content	9
List of Figures	11
List of Tables	12
CHAPTER 1	13
INTRODUCTION	13
1.1 Background	13
1.2 Problem Statement	13
1.3 Objectives	14
1.4 Methodology	14
1.5 AWS Cloud	14
1.6 Web Application	15
1.7 Conclusion	15
CHAPTER 2	16
LITERATURE REVIEW	16
2.1 Literature Review	16
CHAPTER 3	21
METHODOLOGY	21
3.1 Vital Signs:	21
3.2 Data Communication Techniques:	23
3.2.1 Wired Communication:	23
3.2.2 Wireless Communication:	24
3.3 Amazon Web Services:	25
3.4 Web Application:	26
3.5 System Model:	27
3.6 Simulation:	28
3.7 Conclusion:	29
CHAPTER 4	30
MATHEMATICAL MODEL	30
4.1 DS18B20 Digital Temperature Sensor:	30
4.2 MAX30100 Pulse Oximeter Sensor:	32
4.3 AD8232 ECG Sensor:	34
CHAPTER 5	36
CONCLUSION AND RESULTS	36
5.1 Conclusion:	36
5.2 Results:	37
5.3 Hardware:	39
CHAPTER 6	45
BUSINESS MODEL	45
6.1 Introduction:	45
6.2 Problem Identification:	45
6.3 Solution:	45
6.4 Company name and logo:	45
6.5 Product:	45
6.6 SWOT Analysis:	46
6.7 Competition Analysis:	46
6.8 Unique Selling Propositions (USPs) of Project:	46
6.9 Business Model:	47
6.10 Target Market:	47
6.11 Financial Infrastructure:	47
6.12 Team:	48
6.13 Contact:	48

REFERENCES: 49

List of Figures

Figure 1.1: Overview of the Server-less Patient Monitoring System.	13
Figure 1.2: Architecture of state of the art Event-Driven Server-less Patient Monitoring System.	15
Figure 3.1: 3-Lead ECG chart.	21
Figure 3.2: Pulse Oximeter sensor.	22
Figure 3.3: Working flowchart of Pulse Oximeter Sensor.	22
Figure 3.4: Age wise Body Temperature.	22
Figure 3.5: Blood oxygen level chart.	23
Figure 3.6: Data Communication through UART Protocol.	23
Figure 3.7: Data Communication through SPI Protocol.	24
Figure 3.8: Data Communication through I2C Protocol.	24
Figure 3.9: Block Diagram of MQTT Communication Architecture.	25
Figure 3.10: Dynamo DB.	25
Figure 3.11: Lambda (Amazon Web Services).	26
Figure 3.12: AWS Web-Sockets.	26
Figure 3.13: API Gateway for Web Application.	26
Figure 3.14: System Model.	27
Figure 3.15: Circuit Diagram of Patient Monitoring System.	28
Figure 5.1: Proteus Simulation of the Patient Monitoring System.	37
Figure 5.2: Graph of Body Temperature.	37
Figure 5.3: Graph of Pulse Rate.	38
Figure 5.4: Graph of Oxygen level (SpO2).	38
Figure 5.5: Graph of Blood Pressure (diastolic).	38
Figure 5.6: Raspberry Pi 4	39
Figure 5.7: ECG Sensor	40
Figure 5.8: OLED Display	40
Figure 5.9: Analog to Digital Converter	40
Figure 5.10: Digital Temperature Sensor	41
Figure 5.11: Pulse Oximeter Sensor	41
Figure 5.12: Power Supply 5V 2A	41
Figure 5.13: Circuit Diagram	42
Figure 5.14: Raspberry Pi 4 pinout	42
Figure 5.15: Project Hardware.	43
Figure 5.16: Graph of Body Temperature.	43
Figure 5.17: Graph of Pulse Rate.	43
Figure 5.18: Graph of Oxygen level (SpO2).	44
Figure 5.19: ECG waveform	44
Figure 6.1: Startup Logo	45
Figure 6.2: Product overview	45

List of Tables

Table 2.1: Proposed work	18
Table: 3.1 Normal rate of vital signs in the human body.	21
Table 4.1: Analog temperature Data to Digital conversion.	31
Table 5.1: Data on Vital Signs store to CSV file on ThingSpeak Cloud.	38
Table 5.2: Data on Vital Signs store to CSV file on ThingSpeak Cloud.	44
Table 6.1: SWOT Analysis	46
Table 6.2 Financial model of prototype	47
Table 6.3: Contact Details	48

CHAPTER 1

INTRODUCTION

This chapter presents the background of the server-less patient monitoring system in terms of scientific research. Moreover, we discussed the targets of this thesis and our primary contribution and structure used for this work.

1.1 Background

The ICU in hospitals is generally utilized overall and the medical care centers has recently experienced a significant turn of events and interestingly expanded business. The conclusion of sicknesses and different abnormalities in the human body required an actual assessment in the clinic a couple of years prior. Most people needed to stay in the emergency while getting treatment. This expanded the cost of medical care while putting strain on rustic and far-off medical care centers. With the advancement in innovation throughout the long term, checking a patient from far separated places is presently conceivable.

The Internet of Things (IoT) depends on smart and self-configuring nodes (things) interconnected in a dynamic and global network infrastructure. It addresses quite possibly the most disruptive technologies that make the omnipresent and unavoidable computing scene. The Internet of Things usually refers to the real world and easily overlooked details with restricted capacity and processing ability, and the significant issues about dependability, execution, security, and privacy[1]. Cloud computing is the on-request utilization of processing power, capacity, information bases, applications, and any IT resources through the Internet. The most essential method for characterizing the 'Cloud' is that it is a computer found elsewhere that is gotten to through the internet and used here and there. Web services are likewise one more name for what people call the cloud. The cloud contains server computers situated in various areas all over the planet. At the point when we utilize a cloud administration like Amazon Web Services (AWS), we are using the computers having a place with these Cloud Service Providers [2]. Data Acquisition from ICU shelf monitor is a challenging task. The data lines of the Shelf monitor via SPI/UART/I2C communication protocol that is connected to Raspberry Pi to process and sent to the cloud using MQTT protocol.

In the AWS cloud platform, it has to manage incoming data, process it, and store it in a database using Lambda functions. The API from the cloud is accessed by the Web Application by using JavaScript Object Notation. The web application in Python language communicates data to AWS using JSON. The front end of the web application also uses Python.

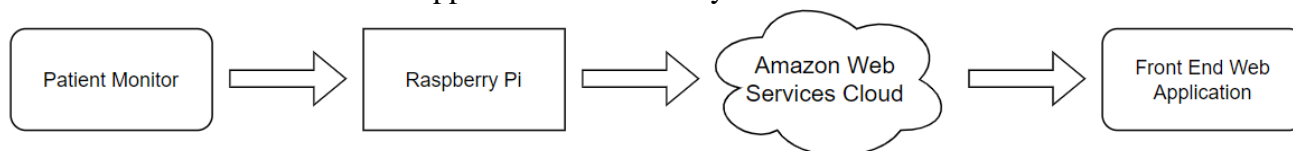


Figure 1.1: Overview of the Server-less Patient Monitoring System.

With the revolution of DAQ architecture in medical machines, the higher data rate from ICU patient monitors to web applications is so fast that every doctor sees the real-time status of the patient.

1.2 Problem Statement

The health system in Pakistan is currently facing numerous challenges. One of them is the limited number of health professionals and their unjustified distribution. This problem is magnified in the case of ICUs, as they are equipped with a large number of patient monitoring systems and a large amount of multidimensional physiological data i.e., heart rate, oxygen saturation, and temperature generated daily.

1.3 Objectives

i. Research Objectives:

- To design a data acquisition system for the ICU patient monitor to acquire physiological vital signs.
- To develop an IoT-based architecture for the data transmission to the designated digital platforms.
- To develop an AI-based mechanism to identify the critical condition of ICU patients.
- Implementation and evaluation of the proposed system.

ii. Industrial Objectives:

- Provide IoT-based smart solutions to hospitals for improvement in the medical care of ICU patients.
- To revolutionize the hospitals with industry 4.0 design principles.
- ML models and big data models are also applied to given data to revolutionize industry 5.0.

1.4 Methodology

The connected sensors of the patient monitor send data to it. The data running on the monitor is sent to Raspberry Pi using a data acquisition method like (SPI/UART/I2C). Data in (Raspberry Pi) is processed and sent to the AWS cloud using the MQTT protocol. AWS has its Data Base, User Authorization, Lambda Functions, and API gateway to transmit data. Web-based Front End of this system communicates to the AWS cloud using JSON data packets.

i. Data Acquisition:

The UART interface does not utilize a clock signal to synchronize the transmitter and receiver devices. It sends data asynchronously. Rather than a clock signal, the transmitter generates a bit stream based on its clock signal while the receiver utilizes its internal clock signal to sample the incoming data.

ii. MQTT Protocol:

MQTT is a messaging protocol that is utilized for machine-to-machine correspondence. Communication between smart sensors and Internet of Things (IoT) gadgets commonly needs to send and get data over a resource-constrained network with limited bandwidth.

1.5 AWS Cloud

i. IoT core:

IoT Core typically refers to a central component or platform in Internet of Things (IoT) systems. IoT Core solutions are designed to connect and manage various devices, sensors, and actuators within the Internet of Things ecosystem.

ii. IoT Rule:

An IoT rule is a set of conditions and corresponding actions that define how the IoT system should react to specific events or data patterns generated by connected devices.

iii. Lambda:

It is part of Amazon Web Services and provides a server-less and event-driven computing platform. The developers can run code without provisioning or managing servers due to its design features.

iv. Quick Sight:

It allows organizations to easily create and publish interactive dashboards that include insights from their data. QuickSight can be used to connect to various data sources, perform analysis, and visualize data through a variety of charts, graphs, and tables.

v. RDS:

It is a managed service by Amazon Web Services (AWS) that makes it easier to manage relational database in the cloud. It provides cost-efficient, resizable capacity while automating time-consuming administration tasks such as hardware provisioning, database setup, patching, and backups.

vi. Dynamo DB:

Amazon DynamoDB is a fully managed NoSQL database service provided by Amazon Web Services (AWS). It offers fast and predictable performance with seamless scalability. It is widely used in industries such as gaming, IoT, mobile, and web applications where scalability, performance, and high availability are critical.

vii. Web Socket:

WebSocket APIs simplify the process of building real-time, bi-directional communication applications. By leveraging AWS Lambda and other AWS services, you can create scalable and efficient WebSocket applications.

viii. API Gateway:

It is a fully managed service provided by AWS (Amazon Web Services) that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. Developers can build robust, scalable, and secure APIs that integrate seamlessly with other AWS services, helping them to focus on building their application logic rather than managing infrastructure.

1.6 Web Application

The web application is a software where the doctors will see the real-time vital signs of their patients that are live for-cast from patient monitor.

Here is the complete block diagram of the project.

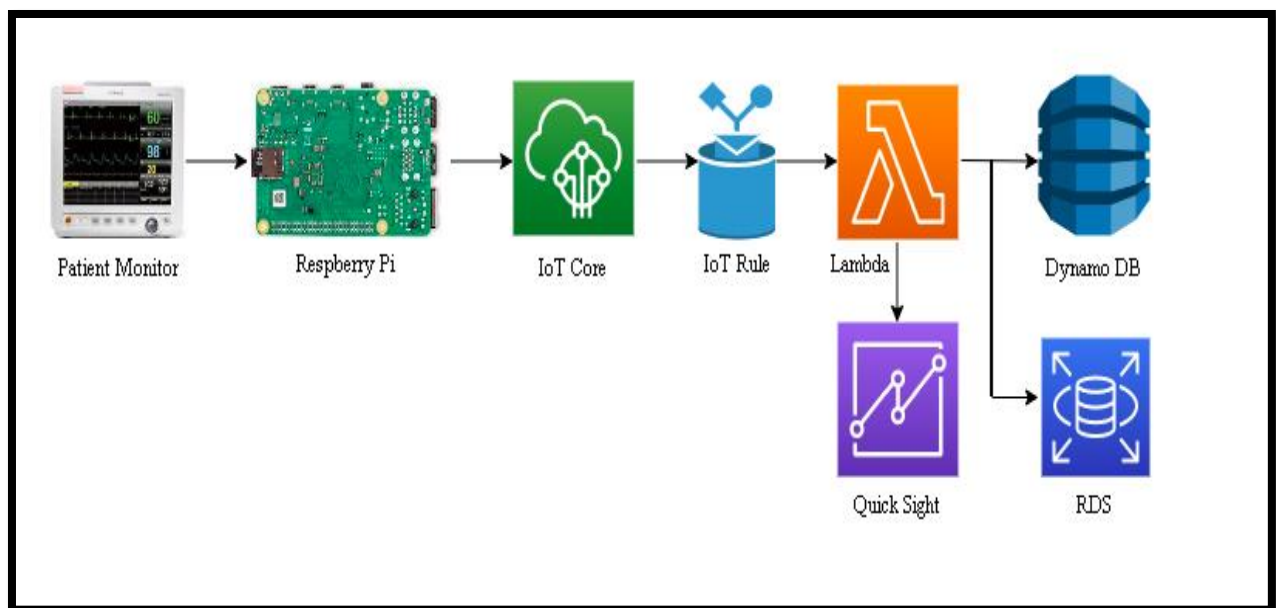


Figure 1.2: Architecture of state of the art Event-Driven Server-less Patient Monitoring System.

1.7 Conclusion

IoT-based event-driven and serverless architectures have become increasingly popular across various industries. Serverless architecture, in particular, has gained prominence in the industrial sector, especially for tasks involving monitoring, automation, and control. Cloud computing enables the on-demand use of computing power, storage, databases, applications, and other IT resources through the Internet. Essentially, the 'Cloud' can be defined as a computer located remotely, accessed via the Internet, and utilized for various purposes.

CHAPTER 2

LITERATURE REVIEW

2.1 Literature Review

i. Data Acquisition (DAQ):

The development of an Android-based mobile data acquisition (DAQ) solution that collects personalized health information from the end user, stores and analyzes the data, and visualizes it on smart devices. Optionally, the data can be sent to a data center for further processing. This smart mobile device can gather information from a wide range of wireless (Bluetooth, Wi-Fi, cloud, and GPRS) and wired (USB) sensors. Additionally, embedded sensors in the mobile device provide valuable status information, such as user location, magnetic or noise levels, acceleration, temperature, and more. Our software solution's user interface is highly configurable, suitable for users of varying skill levels, and includes diary functionality to log information about sleep problems, diet, or even pain. The software also enables correlation analysis between different sensor data sets [1].

ii. Server-less Computing:

Serverless computing is an execution model where the cloud service provider dynamically manages the allocation of compute resources. Consumers are billed based on the actual volume of resources they use, rather than pre-purchasing units of compute capacity. This model emerged to optimize costs, reduce configuration overheads, and enhance the application's scalability in the cloud. Major cloud service providers have embraced the serverless computing model, which is evident in its widespread adoption across various industries [2].

Serverless computing is a modern cloud programming and deployment paradigm gaining widespread popularity. Services like Amazon Web Services (AWS) Lambda, Google Functions, and Azure Functions automatically execute simple functions uploaded by developers in response to cloud-based event triggers. This serverless model greatly simplifies the integration of concurrency and parallelism into cloud applications, enabling the deployment of scalable distributed systems and services at a very low cost [3].

iii. IOT in Healthcare:

The Internet of Things (IoT) has revolutionized digital healthcare by enabling remote monitoring of patients' health conditions. This technology automatically senses and stores patient health data, which is accessible via the internet and alerts doctors to critical conditions [4].

The smart patient health tracking system utilizes sensors to monitor vital signs such as temperature and heartbeat. These sensors are connected to a microcontroller, which interfaces with an LCD and an IoT module for real-time monitoring and alerting. If any sudden changes in the patient's vital signs are detected, the system automatically notifies caregivers via IoT, providing live updates on the patient's heartbeat and temperature. This IoT-based approach effectively leverages internet connectivity to continuously monitor patient health, potentially saving lives through timely intervention [5].

The Internet of Medical Things refers to the connectivity of medical-grade devices equipped with communication capabilities, integrated into broader health networks to enhance patient health. However, due to the crucial nature of healthcare systems, the Internet of Medical Things encounters significant challenges, particularly in ensuring reliability, safety, and security [6].

The evolution of Internet technology enables real-time communication between Internet of Things (IoT) devices and cloud computing systems. This collaborative interaction facilitates seamless healthcare delivery by enabling efficient sharing of patient data and clinical information through a unified networking infrastructure. This makes the Internet an ideal tool for remote patient monitoring applications [7].

In healthcare, IoT applications prove highly beneficial, particularly in remote areas where access to professional medical services is limited due to sparse doctor availability and distance from urban centers. Addressing these challenges, Remote Patient Monitoring emerges as a crucial solution.

Our paper introduces an IoT-based system for real-time patient monitoring. It utilizes MQTT for messaging and facilitates the transmission of Electrocardiogram (ECG) data from a mobile application to a web server, ensuring continuous real-time data monitoring [8].

iv. Smart Health Systems (Wearable and Telemedicine):

Wearable sensors gather comprehensive data, including heart rate and temperature, to assess patient health. Our proposed smart health monitoring system wirelessly monitors and transmits human biosignals in real-time. It comprises three subsystems: one for monitoring heart rate and blood oxygen levels, another for temperature, and a third for detecting falls using wearable sensors. These parameters are then uploaded to the cloud via the Internet of Things (IoT) [9].

An online telemedicine platform fosters collaboration among doctors, hospitals, and patients, facilitating medical consultations for remote areas. Through video calls or text messages, doctors and patients can communicate effectively. Patients can store health information, search for doctors, and consult healthcare professionals via text or video calls. Doctors register on the platform, ensuring authenticity through verification procedure [10].

v. IOMT:

The author acknowledges that while IoMT effectively processes collected data, it faces challenges like data interoperability, standardization, and computational complexity in disease detection. Addressing these concerns, this study explores the application of Iota in patient monitoring systems. Initially, IoMT devices are deployed on the human body to gather health information [11].

vi. Tech for Disease Monitoring:

The systematic literature review (SLR) methodology aims to identify existing technologies designed for monitoring users exhibiting symptoms of COVID-19 and potential future diseases. These technologies are analyzed based on their specific characteristics and categorized according to their intended uses and the types of information they gather [12].

vii. IoT and Cloud in Health Monitoring:

State-of-the-art smart healthcare monitoring integrates communication channels with embedded internal and external sensors, IoT servers, and cloud storage. Health parameters are processed across multiple refining levels: the application layer, management layer, network layer, and device layer. Various sensor data is wirelessly collected from nodes and stored as an unstructured dataset in the cloud [13].

The online telemedicine system facilitates collaboration among doctors, hospitals, and patients, enabling doctors to provide services to patients in remote areas. It supports communication between doctors and patients through video calls or text messages [14].

An Internet-of-Things (IoT) platform was employed for sleep monitoring, integrating contactless or wearable sensors into a cohesive system designed with event-driven and microservice architecture. The system includes multiple services that react to events [15].

viii. Real-Time Monitoring (Key to Disease Detection):

Detecting changes in human heart rate in real-time is crucial for diagnosing this disease. Traditional treatment and prevention methods, however, are unable to achieve real-time monitoring of heart rate changes. Therefore, long-term health monitoring technologies are essential for effective detection and management [16].

ix. Remote Patient Monitoring:

IoT in healthcare integrates diverse medical equipment and sensors with the Internet, facilitating real-time collection and analysis of patient data. Remote patient monitoring (RPM), a key application of IoMT, enables healthcare providers to gather and analyze patient data from a distance. RPM not only decreases the necessity for frequent visits to healthcare facilities but also enhances patient care through e-health solutions, ultimately lowering healthcare costs [17].

x. Wearable Sensor Data Flow:

Wearable sensors positioned on a patient's body gather data, which is initially processed within the wearable device and then wirelessly transmitted to the gateway (RPi). Acting as the gateway edge device, the RPi provides medium-term storage and performs more complex processing tasks. It subsequently sends the data to a cloud computing device either continuously or opportunistically. The cloud device handles long-term data storage, manages comprehensive computational processes, and facilitates interactions with healthcare professionals [18].

xi. Enhancing Accuracy (IoT Patient Monitoring):

A real-time remote patient monitoring system leveraging the Internet of Things (IoT) ensures the precise monitoring of vital signs in real-time. Vital signs are transmitted from the DAQ method to the Raspberry Pi using the Message Queuing Telemetry Transport (MQTT) protocol. This study focuses on reading and analyzing patient vital signs to minimize latency during signal transmission [19].

Table 2.1: Proposed work

Year	Objective	Methodology	Limitations
2023	Health Monitoring System using MQTT protocol [19].	Vital Signs of patients taken by sensors are processed in a microcontroller and transmitted using MQTT Protocol to the mobile application.	It does not have features such as data recovery and database management systems.
2023	Location-based patient health care monitoring system [18].	It is a GPS-based health care system where communication between devices is done by Radiofrequency as well as wifi.	The latency of data will occur because it does not have a cloud system.
2023	Cloud-based monitoring system [17].	Vital signs from sensors are transmitted to the patient monitor wirelessly and it is connected to the cloud. Staff will access data and take action.	The electromagnetic interference against the sensor causes wrong data production from sensors.
2022	Bluetooth based physical Health monitoring system [16].	It is MSP432 based physical health monitoring system using Bluetooth to communicate with smartphones.	It does not contain cloud computing features.
2022	An intelligent sleep quality monitoring system built on an event-driven architecture [15].	It is a patient monitoring system that accesses data of sleeping patients and the data from sensors sent to a device and analyzes the quality of sleep.	It cannot take data like blood pressure, ECG, and oxygen levels.
2021	IOT based healthcare monitoring system during COVID-19 breakout [14].	A robot based patient health care design that can process data on the cloud by Big Data and AI models.	Human senses are not involved in this system to analyze data.
2021	Cloud based patient monitoring system [13].	An Arduino based vital signs collecting system that is sent to Raspberry Pi. Web Application is cloud based that is connected to Data Base.	Raspberry Pi is used as a server that causes data latency.

2021	Evaluating technologies for pandemic surveillance with a focus on safeguarding user privacy [12].	This paper shows the server is to be local or external and also have multi-level authentication.	The system will cause latency due to a separate server.
2021	A patient health monitoring system based on the Internet of Medical Things (IoMT) [11].	This is GPS based positioning and IOT based patient monitoring system in which vital signs are sent to a computer and saved data on database over the internet.	It does not contain a central Database for data communication.
2021	Creating a web-based telemedicine system tailored for COVID-19 patients [10].	Web based telemedicine system that has records of patients, and makes audio as well as video calls with doctors via the internet.	This is appointment based system.
2020	Human Health Monitoring System using Internet of Wearable Things [9].	It is an ESP-32 based IOT patient monitoring system that takes vital signs from patients and sends them to the cloud.	This is a microcontroller based system and the sensor's data is not reliable due to fluctuation.
2020	IOT based remote patient monitoring system [8].	Vital signs are sent to the microcontroller and sent to the database and data is published on a web page.	This is a basic system and cannot be implemented commercially.
2019	A vital signs health monitoring system utilizing IoT [7].	Raspberry Pi based system that takes vital signs from medical sensors and displays them on smartphone.	This cloud cannot handle the data of multiple patients and multiple doctors with better performance.
2018	Contributions focusing on Cyber-Physical Systems in Medicine [6].	IoMT based telemedicine system that ensures QoS in cloud service.	It is a generic model of Telemedicine Architecture.
2018	A system for monitoring health using the Internet of Things [5].	It is a PIC16F877A-based vital signs monitoring system for patients.	This is a challenging task to program IoT in Assembly language.
2018	A patient health monitoring system based on the Internet of Things, utilizing wearable biomedical devices [4].	Arduino-based patient vital sign monitoring and positioning system. Raspberry Pi is used as a server as well as a Database.	It cannot be commercially used because it does not use medically certified sensors (do not follow SOPs).
2018	Tracking causal order in AWS lambda applications [3].	Data of devices sent to AWS Lambda cloud storage	It is a paid service.

2018	Serverless Architecture in cloud computing [2].	Event-based architecture where the server only runs when data is communicating.	Only handled by programmers
2017	The design and implementation of a patient healthcare monitoring system based on IoT and hosted in the cloud [1].	GRPS Architecture interface with ARM-7 in which vital signs are sent to ARM-7.	It does not have internet in this architecture.

CHAPTER 3

METHODOLOGY

3.1 Vital Signs:

Vital signs are clinical measurements that provide essential information about a person's physiological functions. They are used to assess the general health status of an individual and to monitor changes that may indicate medical problems.

Table: 3.1 Normal rate of vital signs in human body.

Sr. No.	Vital Signs	Value
01	ECG.	RR interval = 0.6-1.2 seconds. P wave = 80 milliseconds. PR interval = 120-200 milliseconds.
02	Heart Rate.	60 – 100 beats per mint.
03	Body Temperature.	36.1°C to 37°C. 97.6°F to 99.6°F.
04	Oxygen Level.	95 – 100%.

i. ECG:

The average healthy human heart typically beats at a consistent rate of 60 to 100 beats per minute. Various heart conditions can manifest on an electrocardiogram (ECG), such as irregular heart rhythms (fast or slow), heart defects, coronary artery disease, heart valve issues, or heart enlargement. Normal values for ECG waves and intervals are as follows:

- a) RR interval: 0.6-1.2 seconds.
- b) P wave: 80 milliseconds.
- c) PR interval: 120-200 milliseconds.

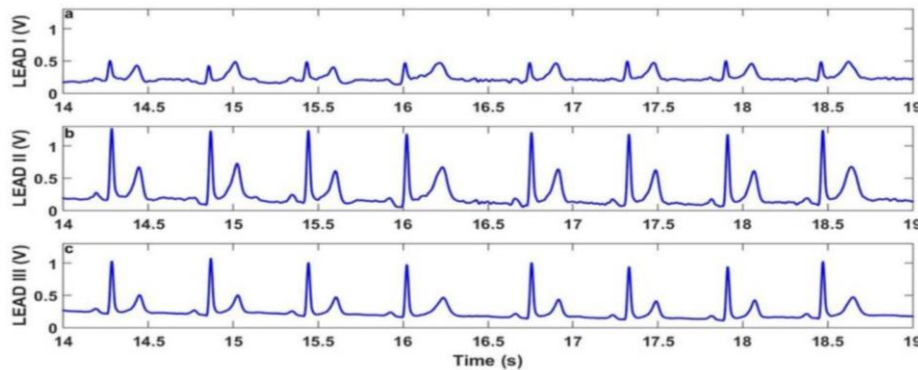


Figure 3.1: 3-Lead ECG chart.

ii. Pulse Rate:

The heart rate, or pulse rate, denotes the frequency of heartbeats per minute (bpm). It indicates both the rhythm and pace of the heart's contractions, offering crucial insights into cardiovascular health. A Reflection-Type Pulse Sensor emits infrared, red, or green light (~550nm) towards the body and gauges the reflected light using a photodiode or phototransistor. This method enables non-invasive measurement of heart rate by detecting changes in light absorption due to blood flow.

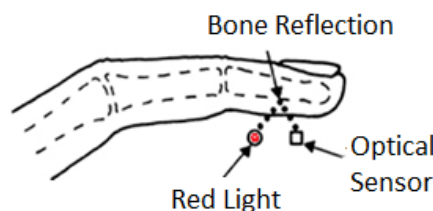


Figure 3.2: Pulse Oximeter sensor.

The pulse oximeter sensor working flow chart is given below:

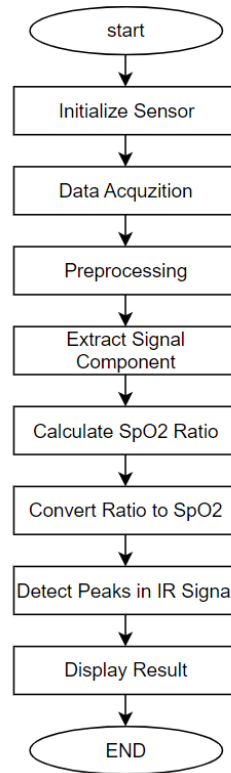


Figure 3.3: Working flowchart of Pulse Oximeter Sensor.

iii. Body Temperature:

The typical body temperature for a healthy adult is generally around 98.6 degrees Fahrenheit (37 degrees Celsius), with slight variations possible. Individual baseline temperatures can differ, influenced by factors such as time of day, age, activity level, and hormonal changes. The accepted average normal body temperature is commonly cited as 98.6°F (37°C), although studies indicate a range from 97°F (36.1°C) to 99°F (37.2°C) as normal. A temperature exceeding 100.4°F (38°C) usually indicates a fever, often resulting from an infection or illness.

Age Group	Oral	Rectal	Core	Ear	Axillary
0 to 2 Years	Not Applicable	97.9 to 100.4°F	97.5 to 100.0°F	97.5 to 100.4°F	94.5 to 99.1°F
3 to 10 Years	95.9 to 99.5°F	97.9 to 100.4°F	97.5 to 100.0°F	97.0 to 100.0°F	96.6 to 98.0°F
11 to 60 Years	97.6 to 99.6°F	98.6 to 100.6°F	98.2 to 100.2°F	96.6 to 99.7°F	95.3 to 98.4°F
60+ Years	96.4 to 98.5°F	97.1 to 99.2°F	96.6 to 98.8°F	96.4 to 99.5°F	96.0 to 97.4°F

Figure 3.4: Age wise Body Temperature.

iv. Oxygen Saturation:

Oxygen saturation in the human body, typically measured as SpO2 (peripheral capillary oxygen saturation), indicates the percentage of hemoglobin in the blood that is saturated with oxygen. It is expressed as a percentage (%). This measurement reflects the proportion of hemoglobin molecules that are currently carrying oxygen compared to those that are not bound to oxygen.

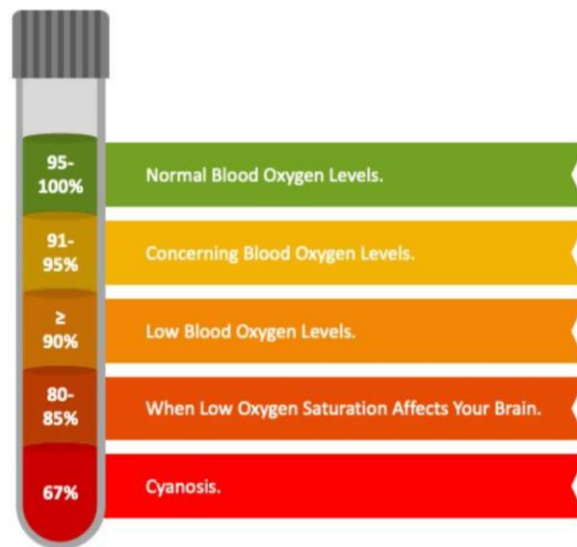


Figure 3.5: Blood oxygen level chart.

3.2 Data Communication Techniques:

Data communication is the digital transfer of data between two points, in other words, data is sent from the sender to the receiver. Telecommunication is the transmission of communication information, such as sound, images, text, etc.

3.2.1 Wired Communication:

A communications cable is used to transmit information using currents of different frequencies. Unlike power cables, which handle high voltages and currents, communication cables are designed for lower voltages and currents.

i. UART communication Protocol:

UART, which stands for universal asynchronous receiver/transmitter, defines protocols for serial data exchange between two devices. It uses a simple two-wire connection between transmitter and receiver, allowing bidirectional communication. UART supports speeds of up to 10 Mbit/s.

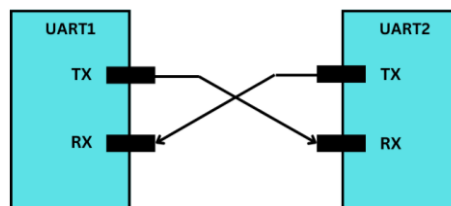


Figure 3.6: Data Communication through UART Protocol.

ii. SPI communication Protocol:

The Serial Peripheral Interface (SPI) facilitates data exchange between multiple devices in a master-slave configuration. The master device initiates and controls communication, while slave devices respond to commands from the master. SPI typically uses four wires: MOSI (Master Out Slave In), MISO (Master In Slave Out), SCK (Serial Clock), and SS (Slave Select). It operates in full duplex mode, allowing simultaneous data transmission and reception. SPI supports various clock speeds, often ranging from a few kilohertz to tens of megahertz, depending on the devices involved. This interface is widely used in embedded systems, microcontrollers, sensors, and other applications requiring fast and reliable data transfer.

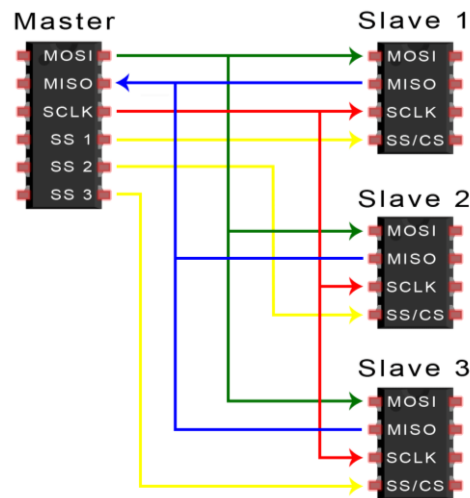


Figure 3.7: Data Communication through SPI Protocol.

iii. I2C Communication Protocol:

It is a serial communication protocol that utilizes two wires: a Serial Data Line (SDA) and a Serial Clock Line (SCL). It enables multiple devices to communicate on the same bus, accommodating various controllers and sensors exchanging data.

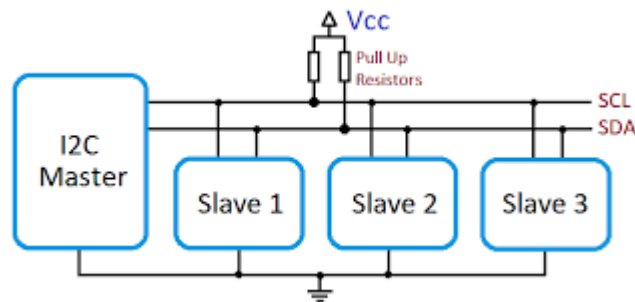


Figure 3.8: Data Communication through I2C Protocol.

3.2.2 Wireless Communication:

i. MQTT Protocol:

Message Queuing Telemetry Transport, is a lightweight messaging protocol designed for scenarios where clients require minimal code footprint and are connected to networks with limited bandwidth or reliability. It facilitates efficient data exchange between constrained devices and server applications, ensuring minimal bandwidth usage and reliability in challenging network conditions. MQTT is favored for machine-to-machine (M2M) communication due to its simplicity of implementation and effectiveness in resource-constrained environments.

There are main three phases of MQTT:

- a. Connect
- b. Publish
- c. Subscribe

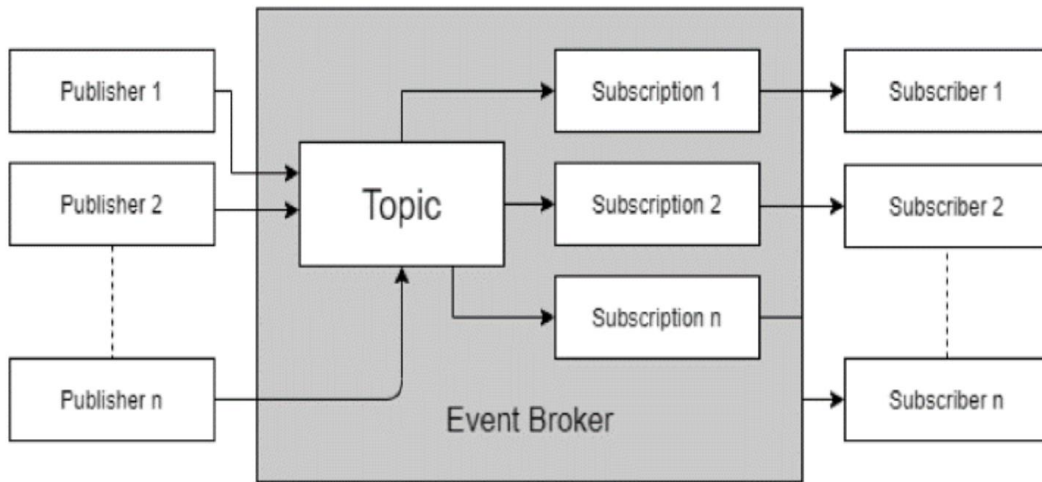


Figure 3.9: Block Diagram of MQTT Communication Architecture.

a) Connect:

MQTT connections are strictly between a client and a broker; clients do not connect directly to each other. To establish a connection, a client sends a CONNECT message to the broker.

b) Publish:

On a private broker, only authorized devices publish and subscribe to topics. This setup is ideal for both production environments and prototyping. Conversely, on a public broker, any device can publish or subscribe to topics hosted on the broker.

c) Subscribe:

MQTT follows a Publish-Subscribe model where a central broker mediates communication. Publishers send messages with specific topics to the broker, which then delivers these messages to subscribers subscribed to those topics, ensuring they receive relevant data.

3.3 Amazon Web Services:

i. Dynamo DB:

Dynamo DB is a fully managed, server-less NoSQL database that excels in handling high-performance applications of any size. It provides built-in security, continuous backups, automated multi-region replication, in-memory caching, and tools for data import and export. Dynamo DB enables customers to manage high-traffic, highly-scalable events effortlessly with nearly limitless throughput. Explore how Amazon Dynamo DB can support various use cases such as content localization, building robust customer data platforms, inventory tracking, and more.



Figure 3.10: Dynamo DB.

ii. Lambda:

Lambda is an event-driven cloud platform provided by Amazon Web Services (AWS), designed for developers to execute code without managing servers. It operates on a high-availability compute infrastructure managed entirely by AWS, handling tasks such as server maintenance, operating system updates, capacity scaling, and logging.



Figure 3.11: Lambda (Amazon Web Services).

iii. Web Socket:

Web Sockets facilitate real-time, event-driven communication between clients and servers, ideal for applications needing instant updates like live chat and messaging. Unlike REST APIs, Web Sockets utilize efficient APIs and maintain a persistent connection to support bidirectional communication in real-time scenarios.



WebSockets

Figure 3.12: AWS Web-Sockets.

iv. API Gateway:

The AWS API Gateway offers numerous advantages, making it indispensable for developers and businesses aiming to create, publish, maintain, monitor, and secure APIs effortlessly, regardless of scale. It seamlessly integrates with various AWS services like AWS Lambda and DynamoDB, enabling the development and deployment of scalable, efficient serverless applications. This integration ensures automatic scaling, allowing APIs to handle high volumes of requests without manual intervention. As a fully managed service, it simplifies the creation, publishing, maintenance, monitoring, and securing of APIs, acting as the gateway for applications to access backend data, business logic, or functionality.

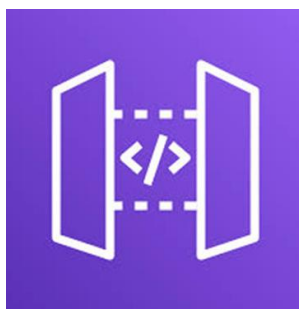


Figure 3.13: API Gateway for Web Application.

3.4 Web Application:

The web application is a software where the doctors will see the real-time vital signs of their patients that are live for-cast from patient monitor.

i. Amazon QuickSight:

Amazon QuickSight is a cloud-based business intelligence service enabling users to create visualizations and dashboards. Competing with tools like Microsoft's Power BI and Tableau, it is fully managed and leverages machine learning for scalable insights creation. QuickSight connects to cloud data sources, combining information from diverse origins via a web-based interface for intuitive data analysis. Users can embed interactive dashboards directly into applications without custom coding, enhancing data visualization and analytics capabilities effortlessly. Utilizing SPICE (Super-fast, Parallel, In-memory Calculation Engine), QuickSight performs complex calculations securely, encrypting data at rest in the Enterprise edition.

3.5 System Model:

This is the model of proposed system.

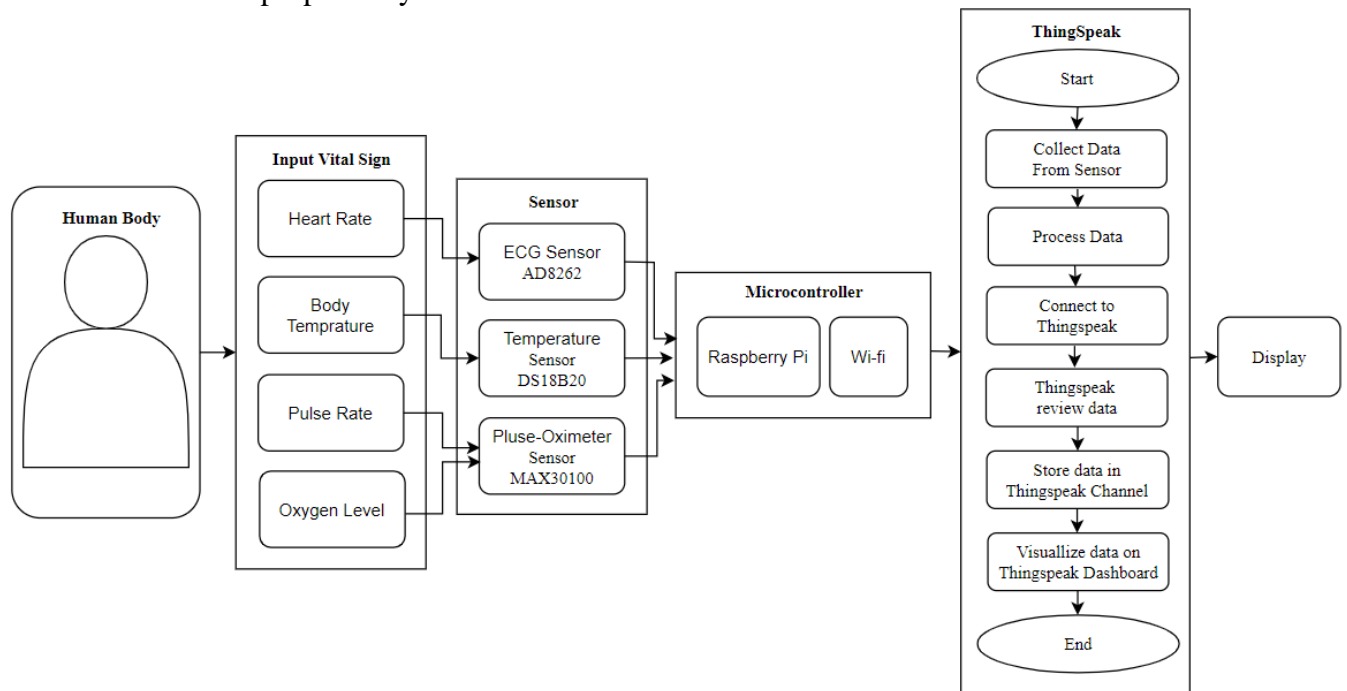


Figure 3.14: System Model.

Following is explanation of each block in the flowchart:

i. Human Body:

The system starts with the human body as the source of vital signs.

ii. Input Vital Signs:

- Heart Rate is the number of heart beats per minute.
- Body Temperature is the measure of the body's thermal state.
- Pulse Rate is the number of pulses per minute, often synonymous with heart rate.
- Oxygen Level is the percentage of oxygen saturation in the blood.

iii. Sensor:

- ECG Sensor (AD8262) is the measures electrical activity of the heart to monitor heart rate.
- Temperature Sensor (DS18B20) is the measures body temperature accurately.
- Pulse-Oximeter Sensor (MAX30100) is the measures pulse rate and oxygen saturation levels in the blood.

iv. Microcontroller:

- Raspberry Pi is a small, affordable computer that serves as the central processing unit, collecting data from the sensors.
- Wi-Fi enables wireless communication, allowing the Raspberry Pi to send data to the cloud (Thing-Speak platform).

v. Thing Speak:

- Start: Initiates the data processing workflow on ThingSpeak.
- Collect Data from Sensor: The Raspberry Pi collects data from the attached sensors (ECG, temperature, pulse-oximeter).
- Process Data: The collected data is processed to make it suitable for transmission and analysis.
- Connect to Thing Speak: The Raspberry Pi establishes a connection to the ThingSpeak platform via the internet.
- Thing Speak Review Data: Thing Speak receives and reviews the incoming data.

- f) Store Data in Thing Speak Channel: The processed data is stored in designated channels on the Thing Speak platform.
- g) Visualize Data on Thing Speak Dashboard: The stored data is visualized using graphs and charts on the Thing Speak dashboard for easy interpretation.
- h) End: Marks the end of the data processing and visualization cycle.

vi. Display:

The final step involves displaying the visualized data on the ThingSpeak dashboard, which can be accessed remotely for monitoring and analysis.

3.6 Simulation:

The simulation of the project and its detail is given bellow:

i. List of Components:

The list of components is given below:

- a) Arduino UNO R3
- b) Serial Communication Port
- c) DHT 11 as (Heart Beat and SpO2 Sensor)
- d) LCD 20x4
- e) LM35
- f) Potentiometer as (Blood Pressure Sensor)
- g) Power Supply

ii. Circuit Diagram:

The circuit Diagram is given below:

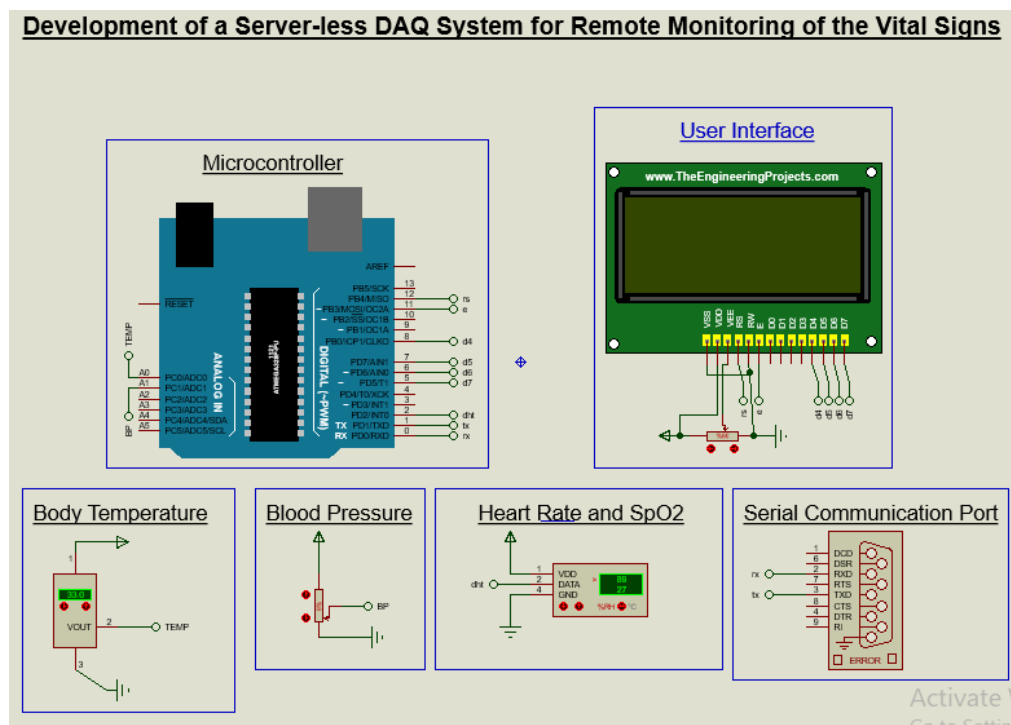


Figure 3.15: Circuit Diagram of Patient Monitoring System.

iii. Connection Sequence:

- a) The Body Temperature sensor signal pin is connected to the A0 pin of Arduino UNO.
- b) The Blood Pressure sensor signal pin is connected to the A0 pin of Arduino UNO.
- c) LCD RS pin to digital pin 12.
- d) LCD E pin to digital pin 11.
- e) LCD D4 pin to digital pin 8.
- f) LCD D5 pin to digital pin 7.

- g) LCD D6 pin to digital pin 6.
- h) LCD D7 pin to digital pin 5.
- i) Heart Beat and SpO2 Sensor pin to digital pin 2.
- j) Digital pins 0 and 1 are the receiver (RX) and transmitter (TX) respectively and these pins are connected to Serial Communication Port.

iv. Arduino Algorithm:

- a) The code starts by including libraries like DHT (humidity and temperature) sensor and LCD display.
- b) Sensor pins are defined for the DHT sensor (D2), body temperature sensor (A0), and blood pressure sensor (A1).
- c) Variables are declared to store readings for heart rate (HR), blood oxygen saturation (O2), body temperature (T), and blood pressure (Bp).
- d) The DHT sensor and LCD are initialized in the setup function, along with setting pin modes for the analog sensors.
- e) The loop function waits for a few seconds and then reads humidity and temperature from the DHT sensor.
- f) Body temperature is measured using an analog sensor and converted to Fahrenheit.
- g) Blood pressure is read from another analog sensor.
- h) The sensor readings are printed to the serial monitor.
- i) A separate function, LCD(), is called to display the sensor readings on a 20x4 LCD screen.
- j) The LCD function first clears the screen and then displays the body temperature, pulse rate, blood oxygen saturation, and blood pressure on it.

3.7 Conclusion:

There are four vital signs that a patient monitor senses i.e. Heart Rate, Blood Pressure, Oxygen Saturation and Body Temperature. The simulation is proof of concept of this project.

CHAPTER 4

MATHEMATICAL MODEL

4.1 DS18B20 Digital Temperature Sensor:

The DS18B20 is a digital temperature sensor commonly used for precise temperature measurement. It communicates over a 1-Wire bus, allowing for easy integration with microcontrollers like Raspberry Pi, and others.

i. Objective:

To develop a mathematical model to estimate the output of a DS18B20 digital temperature sensor at a given temperature of 94°F.

ii. Parameters:

The DS18B20 sensor operates within its specified temperature range and provides accurate readings at the desired temperature. Its resolution is programmable from 9 to 12 bits. It has 1-Wire communication protocol. Multiple sensors can connect on the same data line.

iii. Performance according to parameters:

The DS18B20 digital temperature sensor operates on the principle of converting temperature into a digital signal using its built-in ADC (Analog-to-Digital Converter) and applying a calibration formula to provide accurate temperature readings. The datasheet for the DS18B20 provides a formula to convert the digital output to temperature in Celsius. We can use this formula and then convert the result to Fahrenheit.

The conversion formula provided by the datasheet is:

$$T_{\text{Celsius}} = \frac{9}{5} (T_{\text{Fahrenheit}}) + 32 \quad (1)$$

Where, T_{Celsius} is the temperature in Celsius and T_{Digital} is the digital temperature reading output by the sensor. Given that the sensor operates accurately, we can directly use this formula to calculate the temperature output.

iv. Solution:

Given $T_{94} = 94^\circ \text{F}$ need to convert it to Celsius to use the DS18B20 formula:

$$T_{\text{Celsius}} = \frac{5}{9} (T (\text{digital}) - 32) \quad (2)$$

$$T_{\text{Celsius}} = \frac{5}{9} (98 - 32)$$

$$T_{\text{Celsius}} = \frac{5}{9} (66)$$

$$T_{\text{Celsius}} = 36.67 \quad (3)$$

Now, using the DS18B20 formula to convert T_{Celsius} to Fahrenheit:

$$T_{\text{Fahrenheit}} = \frac{9}{5} (T (\text{celsius}) + 32) \quad (4)$$

$$T_{\text{Fahrenheit}} = \frac{9}{5} (T (36.67) + 32)$$

$$T_{\text{Fahrenheit}} = 66 + 32$$

$$T_{\text{Fahrenheit}} = 98F \quad (5)$$

v. Mathematical Model:

The placement of the DS18B20 sensor is critical for accurate body temperature measurement. Common locations include the underarm, mouth, or skin surface. Placement Adjustment Factor (K_p): Different body locations might have different typical temperature offsets from the core body temperature. For example, under-arm (K_p) is about -0.5 to -1.0°C , mouth (K_p) is about -0.3°C and skin surface varies based on ambient conditions. The raw data

from the DS18B20 sensor is converted to temperature using formula. For 12 bit resolution:

$$T_c = \frac{RAW_{temp}}{16.0} \quad (6)$$

Calibration involves adjusting the raw temperature to match the expected human body temperature range. This requires two main factors, Offset Calibration and Scale Calibration. Offset Calibration (T_{cal}^{scale}) is a constant offset to adjust the baseline temperature and Scale Calibration (T_{cal}^{scale}) adjusts the scaling factor for more accurate conversion. The calibrated temperature (T_{cal}) is:

$$T_{cal} = (T_{raw} T_{cal}^{scale}) + T_{cal}^{offset} \quad (7)$$

Typically, for DS18B20 sensors, the scale calibration is close to 1.0, and the offset calibration is adjusted based on the placement and individual variations. The sensor response time ($T_{response}$) needs to be considered, especially for dynamic measurements where the body temperature changes.

Environmental Adjustment Factor (K_e): Adjust for ambient temperature effects and sensor heat dissipation.

$$T_{cal} = T_{cal} + K_e \quad (8)$$

For instance, if the ambient temperature is significantly different from the body temperature, an additional adjustment might be required.

vi. Example Calculation

Let's put the steps into practice with a hypothetical example:

Assume the DS18B20 reads a raw value of 500. Convert raw data to temperature:

$$T_{raw} = \frac{500}{16.0} = 31.25^\circ\text{C} \quad (9)$$

Assume the sensor is placed under the arm with:

$$T_{cal}^{scale} = 1.0$$

$T_{cal}^{offset} = -0.7^\circ\text{C}$ (adjust for underarm placement)

$$T_{cal} = 31.25 - 0.7 = 30.55^\circ\text{C} \quad (10)$$

Assume a slight adjustment for environmental factors $K_e = 0.1^\circ\text{C}$

$$T_{final} = 30.55 + 0.1 = 30.65^\circ\text{C} \quad (11)$$

vii. Evaluation upon parameters:

The calculated output temperature of the DS18B20 sensor in according from analog value of temperature to digital data is given bellow:

Table 4.1: Analog temperature Data to Digital conversion.

Temperature	Digital Output (Binary)	Digital Output (Hex)
95 °F	0000 0010 0011 0000	0230h
96 °F	0000 0010 0011 1001	0239h
97 °F	0000 0010 0100 0010	0242h
98 °F	0000 0010 0100 1011	024Bh
99 °F	0000 0010 0101 0100	0254h
100 °F	0000 0010 0101 1100	025Ch
101 °F	0000 0010 0110 0101	0265h
102 °F	0000 0010 0110 1110	026Eh
103 °F	0000 0010 0111 0111	0277h
104 °F	0000 0010 1000 0000	0280h

ix. Report:

This process ensures that the DS18B20 sensor provides reliable and accurate measurements when interfaced with the human body. The model was assessed to be valid based on the comparison of the calculated output temperature with the input temperature, confirming its accuracy and reliability for temperature estimation within the specified range of the DS18B20 sensor.

4.2 MAX30100 Pulse Oximeter Sensor:

The MAX30100 is a popular integrated pulse oximetry and heart-rate sensor module designed for wearable health devices. It combines two LEDs, a photodetector, optimized optics, and low-noise analog signal processing to detect pulse oximetry and heart-rate signals. Red LED (660 nm) used for measuring oxygen saturation (SpO₂) levels. Infrared LED (880 nm) used for measuring heart rate and compensating for ambient light interference. The light reflected from the tissues and measures the changes in light intensity due to blood flow, that converts the captured light signal into an analog electrical signal, also includes a low-noise signal processing circuit to enhance the accuracy of the measurements. The module communicates with a microcontroller via the I2C interface, making it easy to integrate with various microcontrollers like Raspberry Pi.

i. Mathematical Model:

Creating a mathematical model for the MAX30100 involves understanding the principles behind pulse oximetry and heart rate detection. The device uses photoplethysmography (PPG), a non-invasive optical technique to detect blood volume changes in the microvascular bed of tissue. Here's a detailed breakdown:

a) Pulse Oximetry Model.

b) Pulse oximetry measures the oxygen saturation (SpO₂) of the blood by comparing the absorption of red and infrared light by oxygenated and deoxygenated hemoglobin.

Beer-Lambert law relates the absorption of light to the properties of the material through which the light is traveling. The intensity of transmitted light through a medium is given by:

$$I = I_0 \cdot e^{-\epsilon c L} \quad (12)$$

Where I is the transmitted light intensity, I_0 is the initial light intensity, ϵ is the molar absorptivity of the absorber, c is the concentration of the absorber, L is the path length of the light through the material. For pulse oximetry, two wavelengths are used. Red light of (660 nm) and Infrared light of (940 nm). Red light absorbed more by deoxygenated hemoglobin. Infrared light absorbed more by oxygenated hemoglobin. The ratio of these absorptions can be used to calculate SpO₂.

The light signal has a pulsatile (AC) component due to arterial blood flow and a non-pulsatile (DC) component due to venous blood, tissue, and bone. The AC component is superimposed on the DC component. The ratio of the AC to DC components at each wavelength is used to compute the ratio of ratios (R):

$$R = \frac{AC_{red}/DC_{red}}{AC_{IR}/DC_{IR}} \quad (13)$$

SpO₂ is then determined using a calibration curve (typically empirically derived):

$$SpO_2 = A - B \cdot R \quad (14)$$

Where, A and B are constants derived from empirical calibration. The AC component of the infrared signal is typically used for heart rate calculation. Peaks in the PPG signal correspond to heartbeats. The time interval between successive peaks (the inter-beat interval, IBI) is measured. Heart rate (HR) is then calculated as:

$$HR = \frac{60}{IBI} \quad (15)$$

Where, IBI is in seconds.

$$I_{red} = I_0 \cdot e^{-\epsilon(\text{red})cL} \quad (16)$$

$$I_{IR} = I_0 \cdot e^{-\epsilon(\text{IR})cL} \quad (17)$$

$$R_{red} = \frac{AC_{red}}{DC_{red}} \quad (18)$$

$$R_{IR} = \frac{AC_{IR}}{DC_{IR}} \quad (19)$$

$$R = \frac{R_{red}}{R_{IR}} \quad (20)$$

Implementation Considerations by Signal Filtering is accurately detect the AC component, filtering out noise and the DC component is crucial. Also, Peak Detection Algorithm accurate the peak detection algorithms are necessary for reliable heart rate measurement. This mathematical model provides the foundation for understanding how the MAX30100 processes signals to derive SpO₂ and heart rate measurements.

Creating a mathematical model of the human body, particularly focusing on the cardiovascular system as it relates to pulse oximetry and heart rate detection, involves several complex physiological processes. The heart pumps blood through the circulatory system, which can be modeled using the cardiac output equation:

$$Q = HR * SV \quad (21)$$

Where Q is the cardiac output (volume of blood pumped by the heart per minute), HR is the heart rate (beats per minute), SV is the stroke volume (volume of blood pumped per beat). Blood flow in arteries can be modeled using the continuity equation and Poiseuille's law:

$$Q = \Delta P \cdot \left(\frac{\pi r^4}{8\eta L} \right) \quad (22)$$

Where ΔP is the pressure difference, r is the radius of the blood vessel, η is the blood viscosity, L is the length of the vessel. Oxygen transport in the blood is primarily carried out by hemoglobin. The oxygen saturation (SpO₂) is the fraction of hemoglobin that is saturated with oxygen.

The oxygen content of arterial blood (CaO_2) can be modeled as:

$$CaO_2 = (Hb \cdot 1.34 \cdot SaO_2) + (0.003 \cdot PaO_2) \quad (23)$$

Where Hb is the hemoglobin concentration, SaO_2 is the arterial oxygen saturation, PaO_2 is the partial pressure of oxygen in arterial blood.

ii. Example Calculation:

Here is a hypothetical example, I_{red} is array of values representing the red light intensity over time and I_{IR} is array of values representing the IR light intensity over time. To calculate AC and DC Components, $DC_{red} = 500$, $AC_{red} = 50$, $DC_{IR} = 600$, $AC_{IR} = 40$. Determine the Ratio of Ratios by using formula,

$$R_{red} = \frac{50}{500} = 0.1$$

$$R_{IR} = \frac{40}{600} = 0.0667$$

$$R = \frac{0.1}{0.0667} = 1.5$$

To calculate SpO₂,

$$SpO_2 = 110 - 25 \cdot 1.5 = 110 - 37.5 = 72.5\%$$

To calculate heart rate, suppose the time interval between successive peaks (IBI) is 0.8 seconds.

$$HR = \frac{60}{0.8} = 75bpm$$

iii. Report:

The mathematical model developed accurately estimates the output readings of a

MAX30100 Pulse oximeter sensor when provided with a pulse rate of 76 bpm and a SpO2 level of 99%. The model was assessed to be valid based on the comparison of the calculated output readings with the input values, confirming its accuracy and reliability for estimating pulse rate and SpO2 level within the specified range of the MAX30100 sensor.

4.3 AD8232 ECG Sensor:

The AD8232 sensor is an integrated signal conditioning block for ECG and other biopotential measurement applications. To model the ECG signal acquisition using the AD8232 sensor, we need to consider the electrode configuration, signal acquisition, amplification and filtering. The AD8232 typically uses a three-electrode configuration (RA, LA, and RL). A (Right Arm), LA (Left Arm), and RL (Right Leg or ground). The heart generates electrical signals that can be detected on the skin surface using electrodes. These signals are weak and need amplification and filtering to be useful. The AD8232 provides an integrated solution for amplifying and filtering the ECG signal. The device includes a two-pole high-pass filter for removing motion artifacts and electrode half-cell potential, and a three-pole low-pass filter for removing high-frequency noise.

i. Mathematical Model:

The raw Input Signal of (ECG) can be modeled as $V_{ECG}(t)$. This signal is typically in the range of 0.5 mV to 5 mV. The high-pass filter removes low-frequency components, including motion artifacts. The transfer function for a high-pass filter can be given by:

$$H_{HPF}(s) = \frac{s}{s + w_c} \quad (24)$$

The amplified signal is $V_{AMP}(t)$ and the amplification factor (gain) is G . Typically, the gain is set to a value that amplifies the ECG signal to a usable range (e.g., 100-1000 times).

$$V_{AMP}(t) = G \cdot V_{HPF}(t) \quad (25)$$

The low-pass filter removes high-frequency noise and the transfer function for a low-pass filter can be given by:

$$H_{LPF}(s) = \frac{w_c}{s + w_c} \quad (26)$$

Here, w_c is the cutoff frequency of the low-pass filter.

ii. Overall Transfer Function:

Combining high-pass filtering, amplification, and low-pass filtering, the overall transfer function $H_{total}(s)$ is:

$$H_{total}(s) = G \cdot \left(\frac{s}{s + w_{HPF}} \right) \cdot \left(\frac{w_{LPF}}{s + w_{LPF}} \right)^3 \quad (27)$$

Here, w_{HPF} and w_{LPF} are the cutoff frequencies of the high-pass and low-pass filters, respectively.

iii. Time-Domain Representation:

To obtain the output ECG signal in the time domain $V_{out}(t)$, you can use the convolution of the input signal $V_{ECG}(t)$ with the impulse response corresponding to the overall transfer function $H_{total}(s)$:

$$V_{out}(t) = V_{ECG}(t) * h_{total}(t) \quad (28)$$

where $h_{total}(t)$ is the inverse Laplace transform of $H_{total}(s)$.

iv. Example Values:

Assume typical values for an AD8232 setup, high-pass filter cutoff frequency (w_{HPF}) is 0.5 Hz, low-pass filter cutoff frequency (w_{LPF}) is 40 Hz and Gain (G) is 1000. In practice, these values would be chosen based on the specific application and requirements of the ECG measurement.

v. Report:

The mathematical model of the ECG signal acquisition using the AD8232 sensor involves high-pass filtering, amplification, and low-pass filtering of the raw ECG signal. The exact values for

the filter cutoff frequencies and the gain will depend on the specific configuration and application of the sensor. This model helps in understanding how the sensor processes the ECG signal and prepares it for further analysis or display.

CHAPTER 5

CONCLUSION AND RESULTS

5.1 Conclusion:

The integration of data acquisition techniques in patient monitoring systems marks a significant advancement in enhancing the efficiency and effectiveness of hospital staff. In contemporary healthcare settings, the increasing demand for medical support, coupled with a dwindling medical workforce, presents a substantial challenge. This disparity is particularly pronounced in developing countries, where the mortality rate in Intensive Care Units (ICUs) has been alarmingly high due to delays in responding to critical conditions.

In response to these challenges, recent technological advancements have paved the way for innovative approaches aimed at transforming healthcare delivery. Among these innovations, the Internet of Things (IoT)-based event-driven and serverless architecture has emerged as a robust solution, gaining widespread adoption across various industries for its efficacy in monitoring, automation, and control.

This project leverages these advancements by implementing data acquisition techniques to monitor patients' vital signs using off-the-shelf patient monitors. The collected data is then transmitted to the cloud platform, which facilitates the communication of this information to subscribers via an API, making it accessible through a smartphone application. The system's frontend is a web-based application that enables doctors to monitor multiple patients' vital signs simultaneously, ensuring timely and efficient medical intervention.

By employing serverless architecture, this project harnesses the benefits of scalability, cost-efficiency, and simplified management, which are critical in handling the dynamic and high-volume data typical of healthcare environments. Serverless computing offloads the complexities of server management, allowing healthcare providers to focus on delivering quality patient care. Additionally, the event-driven nature of this architecture ensures real-time processing and responsiveness, crucial for critical care scenarios.

The implementation of this system in healthcare settings can lead to several key benefits. Firstly, it improves the speed and accuracy of patient monitoring, as vital signs are continuously tracked and anomalies are promptly flagged for medical attention. This real-time monitoring is essential for early detection of potential health deteriorations, enabling proactive interventions that can save lives. Secondly, the system enhances the capacity of healthcare providers to manage multiple patients simultaneously, thereby addressing the issue of staff shortages. With the ability to access patient data remotely, doctors can provide timely advice and interventions, even when they are not physically present in the hospital.

Moreover, the integration of this system with a smartphone application democratizes access to vital health information, making it possible for patients and their families to stay informed about their health status. This transparency fosters a collaborative approach to healthcare, where patients are more engaged in their treatment plans and can make informed decisions about their health. Additionally, the use of cloud computing ensures that patient data is stored securely and can be accessed from anywhere, providing flexibility and continuity of care.

In conclusion, the adoption of IoT-based event-driven and serverless architecture in patient monitoring systems represents a significant leap forward in addressing the challenges faced by modern healthcare systems. By improving the efficiency of medical staff and enabling real-time, remote monitoring of patients, this system has the potential to significantly reduce mortality rates in ICUs, particularly in developing countries. The seamless integration of data acquisition, cloud computing, and mobile technology not only enhances patient care but also opens new avenues for medical research and

public health improvements. As healthcare continues to evolve, such innovative solutions will be instrumental in meeting the growing demands of medical support and ensuring better health outcomes for all.

5.2 Results:

Here is the simulation results:

i. Proteus Simulation Results:

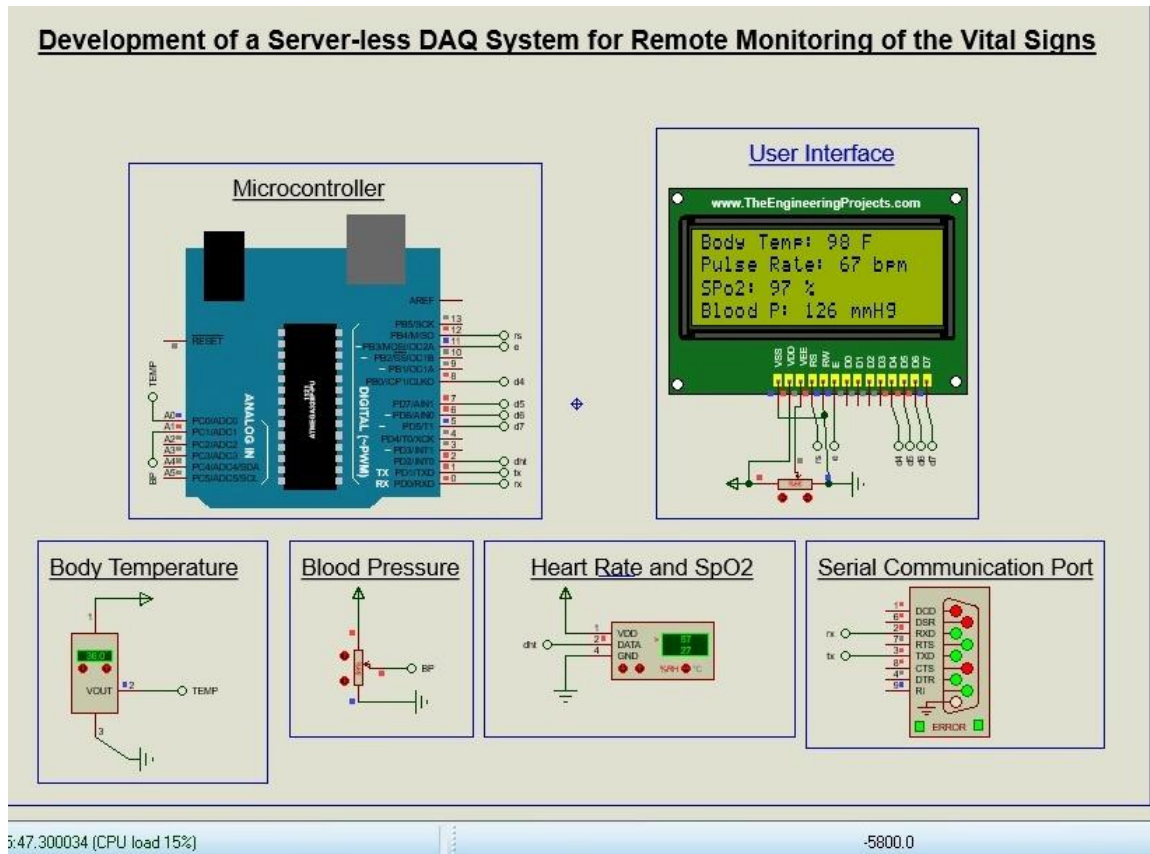


Figure 5.1: Proteus Simulation of the Patient Monitoring System.

ii. ThingSpeak Results:

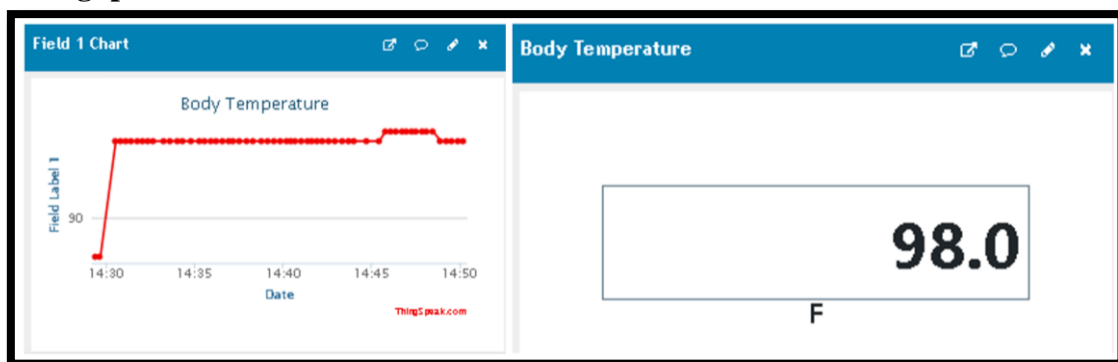


Figure 5.2: Graph of Body Temperature.



Figure 5.3: Graph of Pulse Rate.



Figure 5.4: Graph of Oxygen level (SpO2).

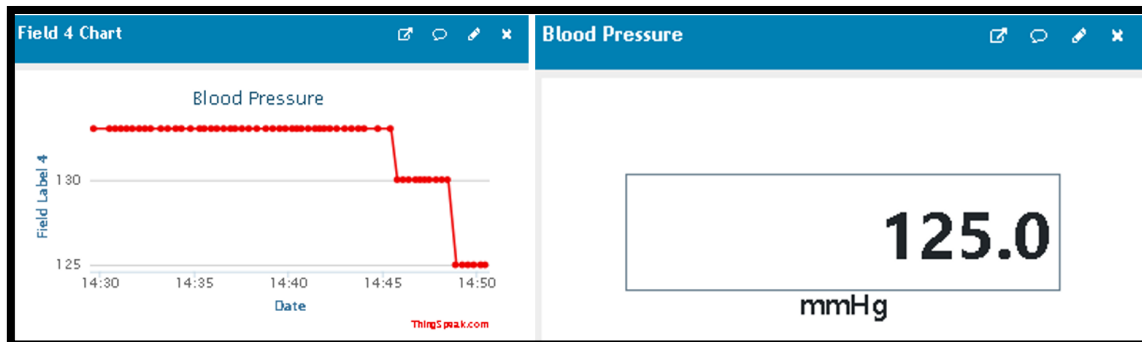


Figure 5.5: Graph of Blood Pressure (diastolic).

Here is the recorded data on CSV file of thingSpeak

Table 5.1: Data on Vital Signs store to CSV file on ThingSpeak Cloud.

Date	Time	Interval	Body Temperature (°F)	Heart Rate (bpm)	SpO2 (%)	Blood Pressure (mmHg)
2024-01-28	02:32:43	1	99	98	73	130
2024-01-28	02:33:01	2	99	98	73	130
2024-01-28	02:33:18	3	99	98	73	130
2024-01-28	02:33:34	4	99	98	73	130
2024-01-28	02:33:50	5	99	98	73	130
2024-01-28	02:34:07	6	99	98	76	130
2024-01-28	02:34:23	7	99	98	76	130
2024-01-28	02:34:39	8	98	99	76	125
2024-01-28	02:34:55	9	98	99	76	125
2024-01-28	02:35:10	10	98	99	76	125

5.3 Hardware:

i. Introduction:

The hardware setup for this thesis is designed to acquire vital sign values using an AD8232 heart rate monitor sensor, MAX30100 pulse oximeter sensor and DS18B20 temperature sensor. This section details the components used, their specifications, and their integration into the overall system.

ii. List of Hardware Components

- a) Raspberry Pi
- b) SD Card for Raspberry Pi
- c) OLED Display
- d) AD8232 Heart Rate Monitor Sensor
- e) Electrodes (RA/LA /RL)
- f) ADS1115 16-bit Analog to Digital converter
- g) DS18B20 Digital Temperature Sensor
- h) MAX30100 Pulse Oximeter Sensor
- i) Breadboard and Connecting Wires
- j) Power Supply (5V)

iii. Detailed Description of Key Components

a) Raspberry Pi

- **Name and Model:** Raspberry Pi 4 Model B
- **Specifications:**
 - Supply Voltage: 5V, 2A
 - RAM: 1 GB

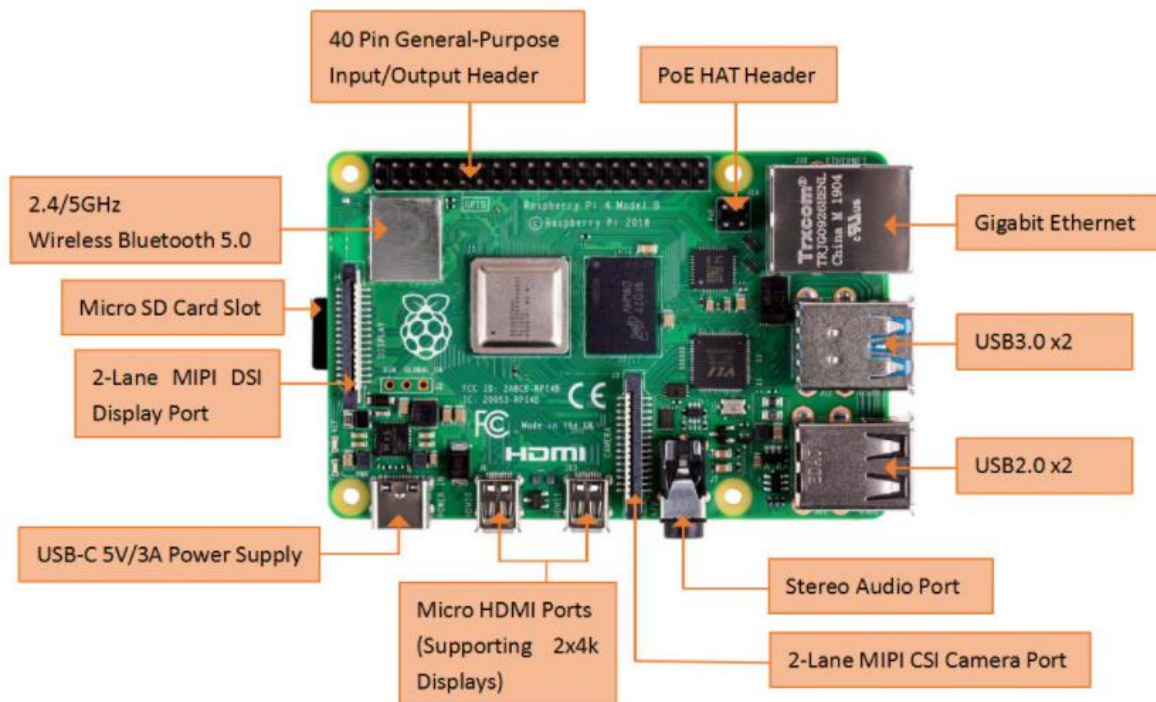


Figure 5.6: Raspberry Pi 4

b) AD8232 Heart Rate Monitor Sensor

- **Name and Model:** Heart Rate Monitor (AD8232)
- **Specifications:**
 - Supply Voltage: 3.3V
 - Data: Analog output

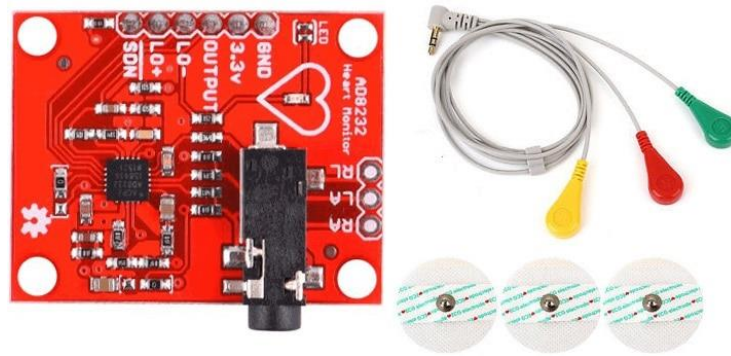


Figure 5.7: ECG Sensor

c) **OLED Display**

- **Name and Model:** 0.96 cm OLED Display Module (SSD1306)
- **Specifications:**
 - Supply Voltage: 5V
 - Data communication protocol: I2C
 - Size: 0.96 cm



Figure 5.8: OLED Display

d) **Analog to Digital Converter**

- **Name and Model:** ADS1115 16-Bit I2C ADC
- **Specifications:**
 - Supply Voltage: 5V
 - Analog pins: A0, A1, A2, A3
 - Data communication protocol: I2C

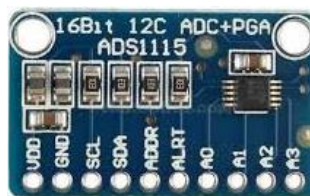


Figure 5.9: Analog to Digital Converter

e) **Digital Temperature Sensor:**

- **Name and Model:** DS18B20 Digital Temperature Sensor
- **Specifications:**
 - Supply Voltage: 5V
 - Data pin: Digital output
 - Data communication protocol: 1-wire bus protocol

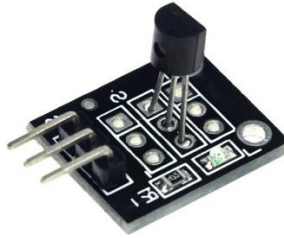


Figure 5.10: Digital Temperature Sensor

f) Pulse Oximeter Sensor:

- **Name and Model:** Pulse Oximeter (MAX30100)
- **Specifications:**
 - Supply Voltage: 5V
 - Data protocol: I2C



Figure 5.11: Pulse Oximeter Sensor

g) Power Supply:

- **Name and Model:** 5V Type-C Adapter
- **Specifications:**
 - Supply Voltage: 5V
 - Current: 2A



Figure 5.12: Power Supply 5V 2A

v. System Architecture:

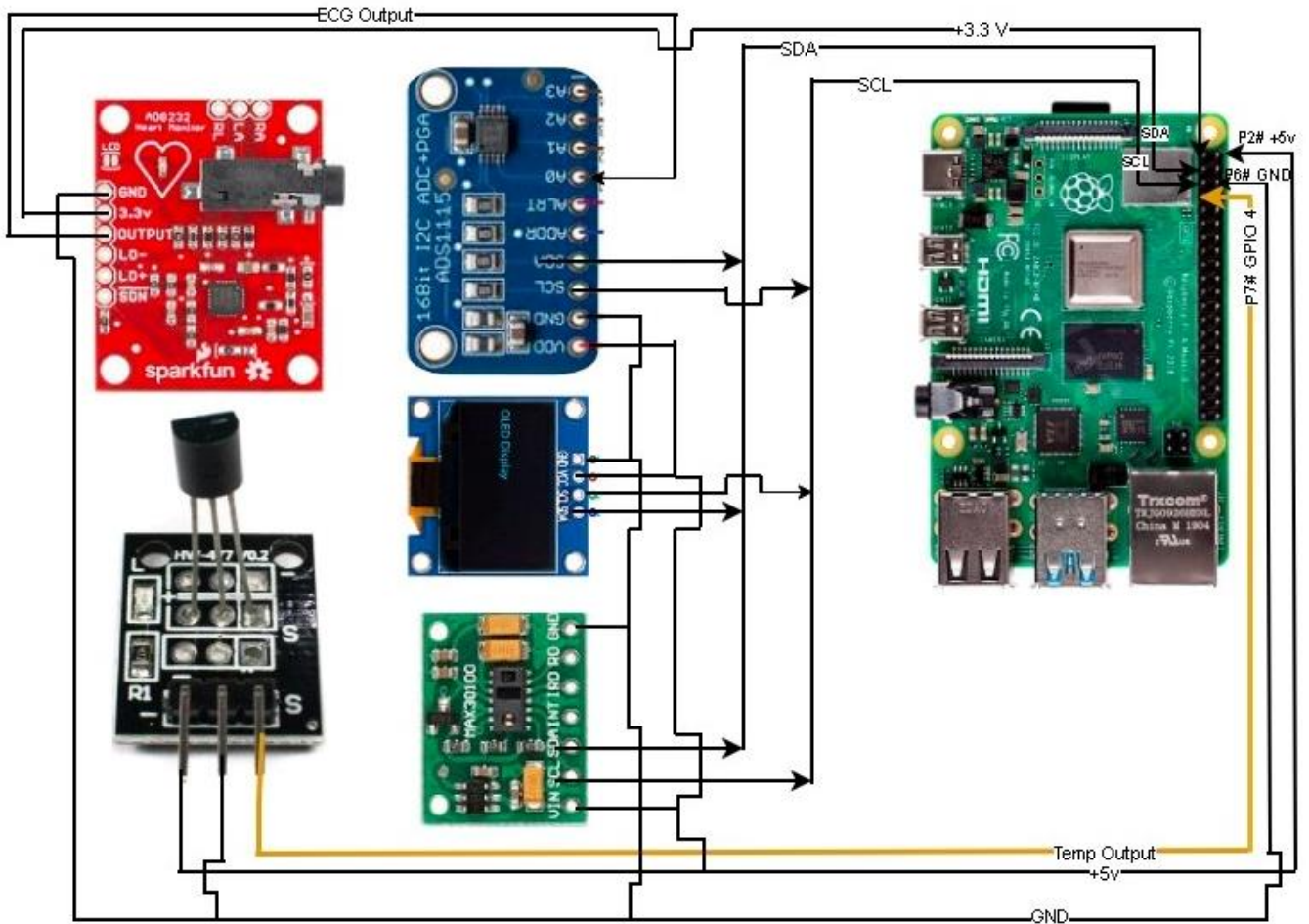


Figure 5.13: Circuit Diagram

vi. Raspberry Pi 4 pinout:

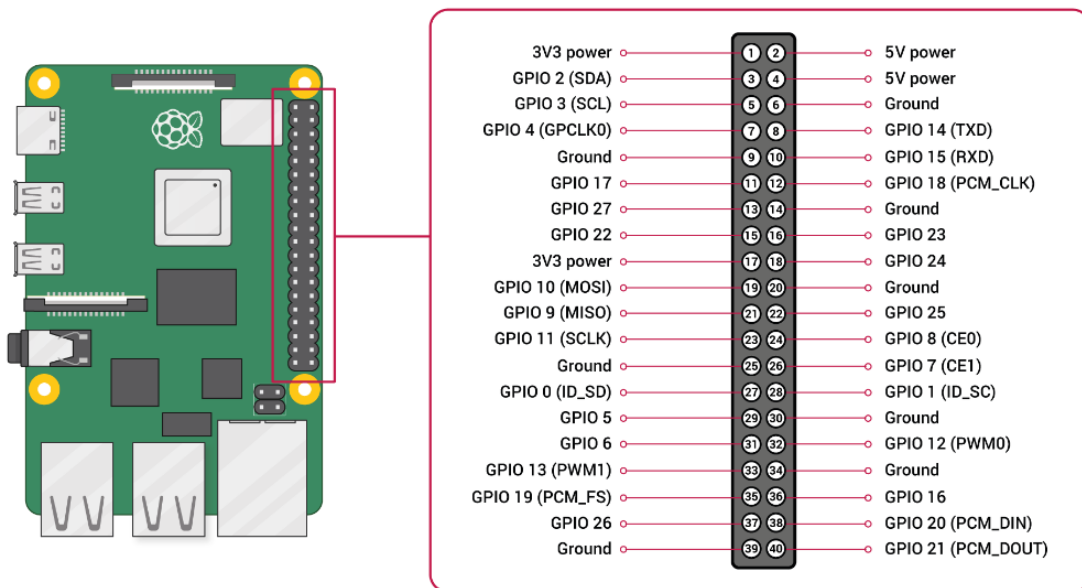


Figure 5.14: Raspberry Pi 4 pinout

viii. Project Hardware:

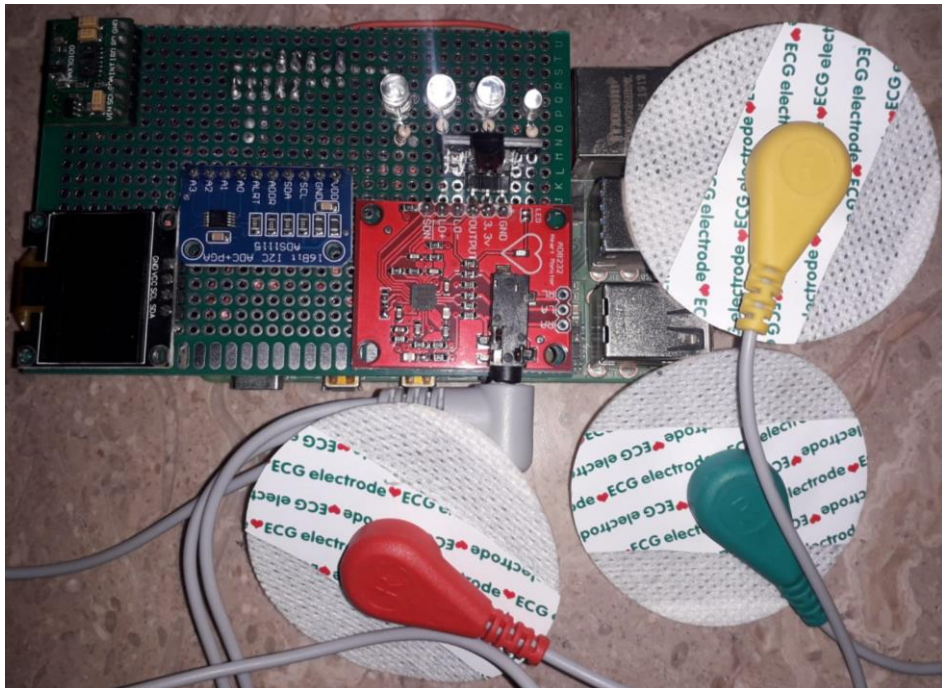


Figure 5.15: Project Hardware.

ix. ThingSpeak Cloud Results:

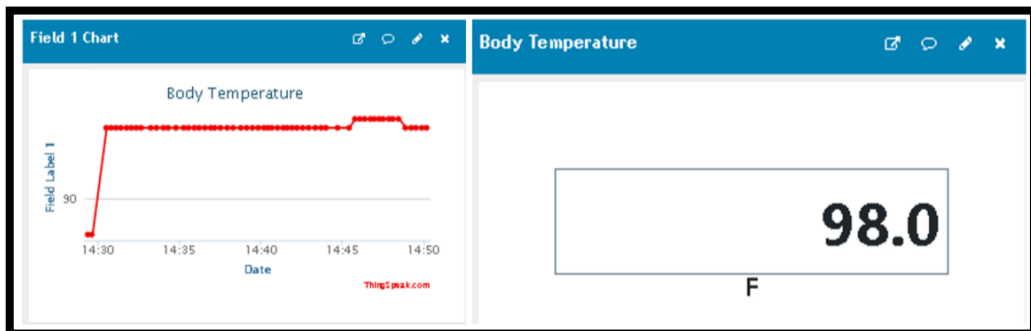


Figure 5.16: Graph of Body Temperature.



Figure 5.17: Graph of Pulse Rate.

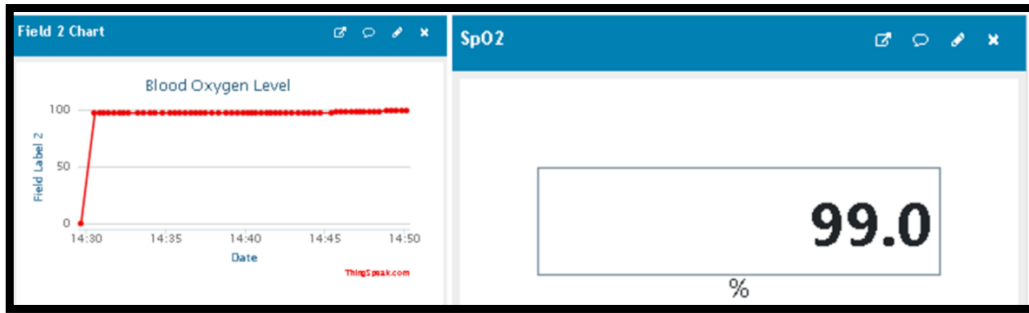


Figure 5.18: Graph of Oxygen level (SpO2).

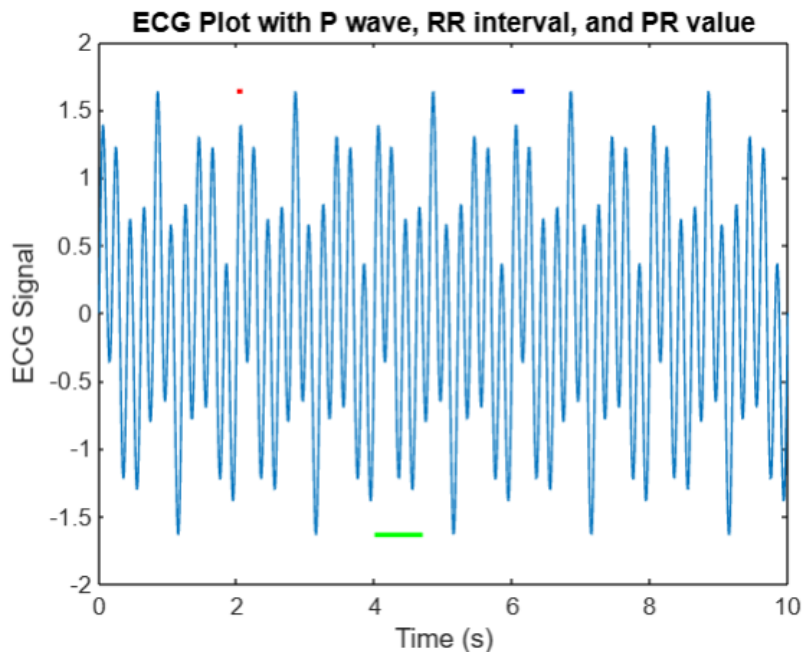


Figure 5.19: ECG waveform

Here is the recorded data on CSV file of thingSpeak.

Table 5.2: Data on Vital Signs store to CSV file on ThingSpeak Cloud.

Date	Time	Interval	Body Temperature (°F)	SpO2 (%)	Pulse Rate (bpm)	RR interval (s)	P wave (ms)	PR interval (ms)
2024-04-05	04:35:43	1	96	98	72	0.7	80	175
2024-04-05	04:36:01	2	96	98	72	0.7	80	175
2024-04-05	04:36:18	3	96	98	72	0.7	80	173
2024-04-05	04:36:34	4	96	98	73	0.8	80	174
2024-04-05	04:36:50	5	96	98	73	0.8	80	174
2024-04-05	04:36:07	6	96	98	73	0.8	80	174
2024-04-05	04:37:23	7	96	98	73	0.7	80	175
2024-04-05	04:37:39	8	96	99	72	0.7	80	175
2024-04-05	04:37:55	9	96	99	72	0.7	80	175
2024-04-05	04:38:10	10	96	99	72	0.7	80	174

CHAPTER 6

BUSINESS MODEL

6.1 Introduction:

MediPiSense is an innovative healthcare technology startup focused on revolutionizing patient monitoring through a serverless, cloud-based solution. Our flagship product leverages the power of Raspberry Pi modules and the ThingSpeak cloud platform to provide real-time monitoring of vital signs such as pulse rate, oxygen level, ECG, and human body temperature.

6.2 Problem Identification:

The health system in Pakistan is currently facing numerous challenges. One of them is the limited number of health professionals and their unjustified distribution. This problem is magnified in the case of ICUs, as they are equipped with a large number of patient monitoring systems and a large amount of multidimensional physiological data i.e., heart rate, oxygen saturation, and, temperature generated daily.

6.3 Solution:

- i. Provide IoT-based smart solutions to hospitals for improvement in the medical care of ICU patients.
- ii. To revolutionize the hospitals with industry 4.0 design principles.

6.4 Company name and logo:



Figure 6.1: Startup Logo

6.5 Product:

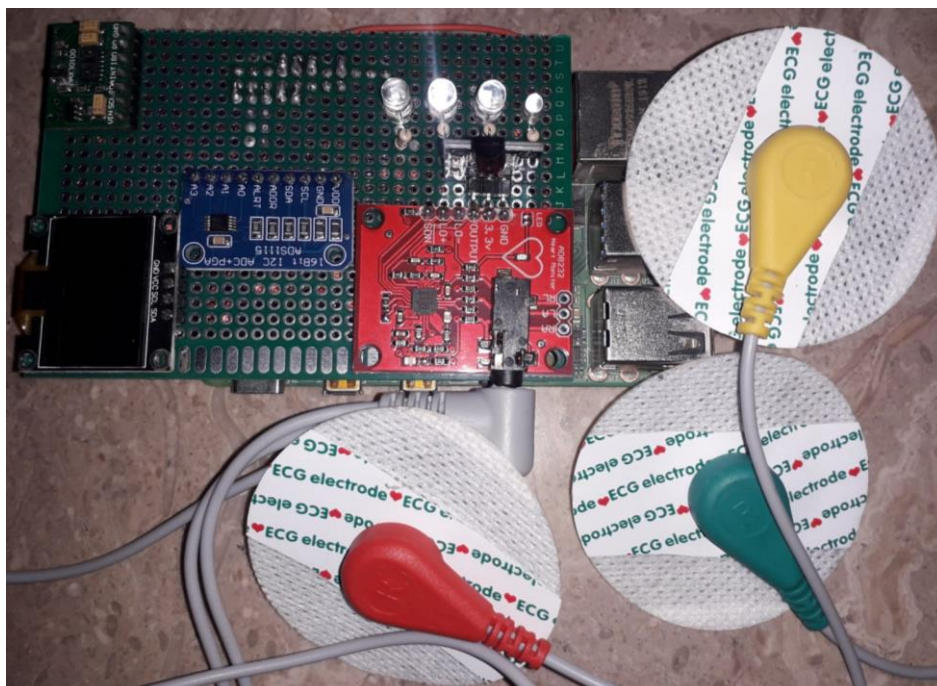


Figure 6.2: Product overview

6.6 SWOT Analysis:

Table 6.1: SWOT Analysis

<p style="text-align: center;">Strengths:</p> <ul style="list-style-type: none"> i. Cost-Effective Solution. ii. Real-Time Monitoring. iii. Scalability. iv. Remote Access. 	<p style="text-align: center;">Opportunities:</p> <ul style="list-style-type: none"> i. Growing Telehealth Market. ii. Expansion to New Markets. iii. Technological Advancements. iv. Healthcare Cost Reduction.
<p style="text-align: center;">Weaknesses:</p> <ul style="list-style-type: none"> i. Reliance on Internet Connectivity. ii. Data Security and Privacy. iii. Technical Complexity. iv. Initial Setup Costs. 	<p style="text-align: center;">Threats:</p> <ul style="list-style-type: none"> i. Regulatory Challenges. ii. Market Competition.

6.7 Competition Analysis:

i. Philips eCareCoordinator

- a) **Description:** Provides a comprehensive remote patient monitoring platform with advanced analytics and telehealth integration.
- b) **Competitive Advantage:** Strong brand, extensive healthcare network.
- c) **Drawback:** Expensive and complex to implement for smaller healthcare providers.

ii. Medtronic Care Management Services

- a) **Description:** Offers a variety of remote patient monitoring solutions focused on chronic disease management.
- b) **Competitive Advantage:** Proven effectiveness in chronic disease management.
- c) **Drawback:** High cost, less flexibility in system customization.

iii. HealthPatch MD (VitalConnect)

- a) **Description:** Wearable biosensor that provides continuous monitoring of multiple vital signs.
- b) **Competitive Advantage:** Advanced sensor technology, highly accurate data.
- c) **Drawback:** Expensive, requires significant investment in hardware.

6.8 Unique Selling Propositions (USPs) of Project:

- i. **Cost-Effectiveness:** Utilizes affordable Raspberry Pi modules, making it accessible to a broader range of healthcare providers and patients.
- ii. **Customizability and Flexibility:** Open-source nature allows for customization to meet specific healthcare needs and integrate with other systems.
- iii. **Real-Time Monitoring with Cloud Integration:** Provides real-time data on vital signs with seamless transmission to the ThingSpeak cloud for analysis and visualization.
- iv. **Ease of Deployment:** Simplifies the deployment process with serverless architecture, reducing the need for extensive IT infrastructure.
- v. **Scalability:** Can scale easily to monitor numerous patients simultaneously, suitable for both small clinics and large hospitals.
- vi. **Remote Access and Monitoring:** Enables healthcare professionals to monitor patients remotely, improving healthcare accessibility in rural and underserved areas.

6.9 Business Model:

i. Customer Segments:

Customer Segments is (B2B) and (B2C).

- a) **Healthcare Providers:** Hospitals, clinics, and home healthcare services looking for cost-effective patient monitoring solutions.
- b) **Telehealth Services:** Telehealth platforms seeking to enhance their remote monitoring capabilities.
- c) **Government and Non-Profits:** Organizations focused on improving healthcare access in rural and underserved areas.
- d) **Senior Care Facilities:** Institutions providing care for elderly patients requiring continuous health monitoring.
- e) **Individual Consumers:** Patients and families interested in personal health monitoring devices for home use.

ii. Revenue Streams:

- a) **Subscription Model:** Monthly or annual subscription fees for accessing the monitoring platform and data analytics services.
- b) **Hardware Sales:** Sale of Raspberry Pi kits pre-configured with sensors and software.
- c) **Software Licensing:** Licensing fees for healthcare providers to integrate the monitoring software with their existing systems.
- d) **Service and Support:** Fees for installation, setup, and training services. Also, Ongoing technical support and maintenance contracts.

6.10 Target Market:

- i. Healthcare Providers
- ii. Telehealth Services
- iii. Senior Care Facilities
- iv. Government and Non-Profit organization
- v. Individual Consumers
- vi. Insurance Companies
- vii. Research Institutions

6.11 Financial Infrastructure:

Table 6.2 Financial model of prototype

Sr no.	Components	Price/Piece	Qty.	Cost
1	Raspberry Pi 4	30,000	1	30,000
2	Adapter 5V 2A	450	1	450
3	AD8232 ECG sensor	1,700	1	1,700
4	ADS1115 ADC	800	1	800
5	DS18B20 temperature sensor	250	1	250
6	OLED display	600	1	600
7	PCB Board	800	1	800
	Total			34,600

6.12 Team:

There are four team members:

- i. Sohail Anjum
- ii. Hamad Rafiq
- iii. Ali Hammas
- iv. M. Samiullah

6.13 Contact:

Table 6.3: Contact Details

Sr.	Name	E-mail	Phone no.
1.	Sohail Anjum	Beem-f20-008@superior.edu.pk	+92 315 6461254
2.	Hamad Rafiq	hamadrafiq0987@gmail.com	+92 320 8425610
3.	Ali Hammas	Beem-f20-019@superior.edu.pk	+92 309 0832633
4.	Muhammad Samiullah	Beem-f20-029@superior.edu.pk	+92 315 4922939

REFERENCES:

- [1] Lakshmanachari, S., Srihari, C., Sudhakar, A., & Nalajala, P. (2017, August). Design and implementation of cloud based patient health care monitoring systems using IoT. In *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)* (pp. 3713-3717). IEEE.
- [2] Rajan, R. A. P. (2018, December). Serverless architecture-a revolution in cloud computing. In *2018 Tenth International Conference on Advanced Computing (ICoAC)* (pp. 88-93). IEEE.
- [3] Lin, W. T., Krintz, C., Wolski, R., Zhang, M., Cai, X., Li, T., & Xu, W. (2018, April). Tracking causal order in aws lambda applications. In *2018 IEEE international conference on cloud engineering (IC2E)* (pp. 50-60). IEEE.
- [4] Ahmed, Z. U., Mortuza, M. G., Uddin, M. J., Kabir, M. H., Mahiuddin, M., & Hoque, M. J. (2018, December). Internet of Things based patient health monitoring system using wearable biomedical device. In *2018 international conference on innovation in engineering and technology (ICIET)* (pp. 1-5). IEEE.
- [5] R Sangeetha, R. S., Jegadeesan, D., Ramya, M. P. ., & Vennila, G. . (2018). HEALTH MONITORING SYSTEM USING INTERNET OF THINGS . *International Journal of Engineering Research and Advanced Technology (ijerat) (E-ISSN 2454-6135)DOI: 10.31695/IJERAT, 4(3), 13–18.* <https://doi.org/10.7324/IJERAT.2018.3201>
- [6] Gatouillat, A., Badr, Y., Massot, B., & Sejdić, E. (2018). Internet of medical things:A review of recent contributions dealing with cyber-physical systems in medicine. *IEEE internet of things journal*, 5(5), 3810-3822.
- [7] Swaroop, K. N., Chandu, K., Gorrepotu, R., & Deb, S. (2019). A health monitoring system for vital signs using IoT. *Internet of Things*, 5, 116-129.
- [8] Maqbool, S., Iqbal, M. W., Naqvi, M. R., Arif, K. S., Ahmed, M., & Arif, M. (2020, November). IoT based remote patient monitoring system. In *2020 International Conference on Decision Aid Sciences and Application (DASA)* (pp. 1255-1260). IEEE.
- [9] Baby, B. E., Kadam, S. B., & Aravindhan, A. (2020). Human health monitoring system using internet of wearable things. In *Emerging Technologies for Sustainability* (pp. 253-259). CRC Press.
- [10] Antor, M. B., Jamil, A. H. M., Mamtaz, M., Khan, M. M., Alshamrani, S. S., & Masud, M. (2021). Development of a Web-Based Telemedicine System for Covid-19 Patients. *Intelligent Automation & Soft Computing*, 30(3).
- [11] Tarish, H. A. (2021). SS-FD: Internet of medical things based patient health monitoring system. *Periodicals of Engineering and Natural Sciences*, 9(3), 641-651.
- [12] Ribeiro-Navarrete, S., Saura, J. R., & Palacios-Marqués, D. (2021). Towards a new era of mass data collection: Assessing pandemic surveillance technologies to preserve user privacy. *Technological Forecasting and Social Change*, 167, 120681.

- [13] Lakshmi, G. J., Ghonge, M., & Obaid, A. J. (2021). Cloud based iot smart healthcare system for remote patient monitoring. *EAI Endorsed Transactions on Pervasive Health and Technology*, 7(28), e4-e4.
- [14] de Morais Barroca Filho, I., Aquino, G., Malaquias, R. S., Girão, G., & Melo, S. R. M. (2021). An IoT-based healthcare platform for patients in ICU beds during the COVID-19 outbreak. *Ieee Access*, 9, 27262-27277.
- [15] Surantha, N., Utomo, O. K., Lionel, E. M., Gozali, I. D., & Isa, S. M. (2022). Intelligent sleep monitoring system based on microservices and event-driven architecture. *IEEE Access*, 10, 42069-42080.
- [16] Lou, J., Yang, F., & Lv, C. (2022). Detection and Application of Wearable Devices Based on Internet of Things in Human Physical Health. *Computational Intelligence and Neuroscience*, 2022.
- [17] Khan, M. A., Din, I. U., Kim, B. S., & Almogren, A. (2023). Visualization of Remote Patient Monitoring System Based on Internet of Medical Things. *Sustainability*, 15(10), 8120.
- [18] Al-Naime, K., Al-Anbuky, A., & Mawston, G. (2023). Internet of Things Gateway Edge for Movement Monitoring in a Smart Healthcare System. *Electronics*, 12(16), 3449.
- [19] Alshammari, H. H. (2023). The internet of things healthcare monitoring system based on MQTT protocol. *Alexandria Engineering Journal*, 69, 275-2