HEALTH MONITORING SYSTEM

A Project Report

Submitted by

SANJANA.B	**(211521205133)**
SARA PRICILLA.A	**(211521205138)**
SWETHA.E.V	**(211521205163)**

in partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

PANIMALAR INSTITUTE OF TECHNOLOGY

ANNA UNIVERSITY : CHENNAI 600 025

NOV-DEC 2023

PANIMALAR INSTITUTE OF TECHNOLOGY
ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report titled **"HEALTH MONITORING SYSTEM"** is the bonafide work of "**SANJANA.B(211521205133), SARA PRICILLA.A(211521205138),** and **SWETHA.E.V(211521205163)"** who carried out the project work under my supervision.

SIGNATURE

Dr.S. Suma Christal Mary M.E.,Ph.D.,
PROFESSOR & HEAD OF THE DEPARTMENT
Department of Information Technology,
Panimalar Institute of Technology,
Poonamallee, Chennai 600123.

SIGNATURE

Ms.R.Suseendra M.Tech.,
ASSISTANT PROFESSOR
Department of Information Technology,
Panimalar Institute of Technology,
Poonamallee, Chennai 600123.

Certified that the candidates were examined in the university project viva-voce Examination held on__________________at Panimalar Institute of Technology, Chennai 600123.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

A project of this magnitude and nature requires kind co-operation and support from many, for successful completion. We wish to express our sincere thanks to all those who were involved in the completion of this project.

We seek the blessing from the **Founder** of our institution **Dr. A. JEPPIAAR, M.A., Ph.D.,** for having been a role model who has been our source of inspiration behind our success in education in his premier institution.Our sincere thanks to the Honorable **Chairman** of our prestigious institution **Mrs. REMIBAI JEPPIAAR** for her sincere endeavor in educating us in her premierinstitution.

We would like to express our deep gratitude to our beloved **Secretary and Correspondent Dr. P. CHINNADURAI, M.A., Ph.D.,** for his kind words and enthusiastic motivation which inspired us a lot in completing this project.

We also express our sincere thanks and gratitude to our dynamic **Directors Mrs.C. VIJAYARAJESHWARI, Dr. C. SAKTHI KUMAR, M.E., Ph.D., and Dr. S. SARANYA SREE SAKTHI KUMAR, MBA., Ph.D.,** for providing us with necessaryfacilities for completion of this project.

We also express our appreciation and gratefulness to our respected **Principal Dr. T. JAYANTHY, M.E., Ph.D.,** who helped us in the completion of the project. We wish to convey our thanks and gratitude to our **Head of the Department, Dr.S.Suma Christal Mary, M.E., Ph.D.,** for her full support by providing ample time to complete our project. Special thanks to our Project Guide **Ms. R.Suseendra, M.Tech., Assistant professor** for their expert advice, valuable information and guidance throughout the completion of the project.

Last, we thank our parents and friends for providing their extensive moral supportand encouragement during the course of the project.

CHAPTER 1

ABSTARCT

Electrocardiogram (ECG) monitoring plays a vital role in healthcare by providing real-time information about a person's cardiac health. In this project, we present an ECG monitoring system that utilizes the ESP32 microcontroller and the AD8232 ECG sensor. The system is designed to be portable, cost-effective, and easy to use, making it suitable for personal health monitoring and medical applications.

In this study, a wireless ECG monitoring system based on the AD8232 signal conditioning module was proposed for real-time signal collecting. The AD8232 ECG Sensor will be connected to the ESP32. Then, by connecting ECG leads to the chest or hand, we'll obtain anECG signal. We'll transfer the ECG graph to the cloud using MQTT Broker utilising Ubidots parameters like API Key or Token. Additionally, we highlight the necessity of smart monitoring systems that control emerging technologies such as deep learning, artificial intelligence (AI), Big Data, and the Internet of Things (IoT) in order to deliver efficient, cost-aware, and completely linked monitoring systems.

The AD8232 sensor is responsible for acquiring ECG signals from the body, while the ESP32 microcontroller processes, analyzes, and transmits the data wirelessly to a remote display or smartphone via Bluetooth or Wi-Fi connectivity. The system is complemented by a user-friendly interface for real-time ECG waveform visualization, heart rate calculation, and data storage.

This project aims to enhance remote patient monitoring, promote early detection of cardiac anomalies, and improve overall healthcare management. The integration of ESP32 and AD8232 presents a promising solution for continuous and non-invasive ECG monitoring, which can be invaluable in preventing and managing cardiac-related disorders.

Keywords:- *Health care, Cardiovascular, ECG, IoT*

TABLE OF CONTENTS

CHAPTER 2

INTRODUCTION

In an era marked by rapid advancements in healthcare technology, the demand for innovative solutions that enhance patient care and monitoring is ever-increasing. The integration of Internet of Things (IoT) devices and sensors into healthcare systems has paved the way for more accessible and accurate health monitoring. This project report presents a comprehensive exploration of an ECG (Electrocardiogram) health monitoring system, developed using the ESP32 module and the AD8232 sensor, which promises to revolutionize the way we monitor and manage cardiac health.Cardiovascular diseases remain a significant global health concern, with heart-related ailments being one of the leading causes of mortality worldwide. Early detection and continuous monitoring of heart conditions are essential to mitigate risks and ensure prompt medical interventions. The ECG, a fundamental diagnostic tool in cardiology, records the electrical activity of the heart, enabling healthcare professionals to assess its rhythm and detect anomalies. By integrating the ESP32 module and the AD8232 sensor, our project aims to make ECG monitoring more accessible, cost-effective, and user-friendly.

The ESP32 module, known for its robust wireless capabilities, and the AD8232 sensor, a specialized ECG sensor, are at the core of this system. The ESP32 not only collects ECG data but also allows for real-time transmission to healthcare providers or caregivers, facilitating remote monitoring. Moreover, the ESP32 can store ECG data for offline analysis, making it a versatile tool for patients and healthcare professionals alike.In the following sections of this report, we will delve into the project's objectives, design and implementation, data acquisition, transmission, and analysis methods, as well as the potential impact of this ECG health monitoring system on healthcare practices. We will also discuss the challenges faced during development and possible future enhancements to the system.

This project report showcases the synergy between cutting-edge technology and healthcare, with the aim of improving the quality of life for individuals prone to cardiac conditions. It underscores the importance of continuous monitoring and early detection, demonstrating how technology can empower individuals to take charge of their health and provide healthcare professionals with valuable insights to deliver timely, efficient care.

The ECG health monitoring system using the ESP32 module and AD8232 sensor represents a step forward in the realm of cardiac health, offering a scalable and accessible solution to a critical healthcare need. By combining the power of IoT with specialized sensors, we pave the way for a future where proactive and personalized cardiac health management is within reach for all.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXIXTING SYSTEM:

There are many existing models for ECG monitoring, but some of the most common include:

Traditional ECG monitoring systems: These systems use electrodes placed on the chest to record the electrical activity of the heart. The data is then transmitted to a monitor, which displays the ECG waveform. Traditional ECG monitoring systems are typically used in hospitals and other clinical settings, but they can also be used for home monitoring.

Wearable ECG monitors: Wearable ECG monitors are small, portable devices that can be worn on the body to track heart activity continuously. They are typically worn on the chest or wrist and use electrodes to record the ECG signal. Wearable ECG monitors are becoming increasingly popular for home monitoring and for tracking fitness levels.

AI-assisted ECG monitoring systems: AI-assisted ECG monitoring systems use artificial intelligence to analyze ECG data and identify abnormalities. These systems can be used to screen for heart disease, monitor patients with existing heart conditions, and detect arrhythmias. AI-assisted ECG monitoring systems are still under development, but they have the potential to revolutionize the way that ECG monitoring is performed.

Here are some specific examples of existing models for ECG monitoring:

AliveCor KardiaMobile: The KardiaMobile is a wearable ECG monitor that can be attached to a smartphone or tablet. It records single-lead ECG readings that can be used to detect arrhythmias, such as atrial fibrillation.

Omron HeartGuide: The HeartGuide is another wearable ECG monitor that attaches to a smartphone. It records single-lead ECG readings and provides real-time feedback on heart rate and rhythm.

Withings ScanWatch: The ScanWatch is a smartwatch that includes an ECG sensor. It can record single-lead ECG readings and monitor heart rate, rhythm, and sleep quality.

Medtronic Reveal LINQ: The Reveal LINQ is an implantable ECG monitor that is inserted under the skin. It records continuously and stores up to three months of data.

BioTelemetry iRhythm Zio XT: The iRhythm Zio XT is a wearable ECG patch that is worn on the chest for up to 14 days. It records continuously and stores up to 14 days of data.

AI-assisted ECG monitoring systems are also becoming increasingly available. For example, the AliveCor KardiaMobile and the Medtronic Reveal LINQ both offer AI-powered features that can help users to interpret their ECG data and detect abnormalities.

In addition to these commercial models, there are also a number of open-source and research-grade ECG monitoring systems available. These systems can be used to develop new ECG monitoring algorithms and to evaluate the performance of existing algorithms.

Overall, there is a wide range of existing models for ECG monitoring available, from traditional ECG monitoring systems to wearable ECG monitors to AI-assisted ECG monitoring systems. The best model for a particular individual will depend on their specific needs and preferences.

3.2 PROPOSED SYSTEM:

We suggested a system that comprises of an ECG sensor that measures an ECG signal from the patient using electrodes linked to the ECG board through normal electrode wires.
The data was transferred to the ESP32 controller, which processed it before sending it to the IoT Cloud Platform. The data was then transmitted through WIFI to the user's phone. We'll need an IoT platform to publish the data to the IoT Cloud. So Ubidots is however one platform that provides a platform for programmers to capture sensor data and transform it into usable information. The Ubidots platform allows any Internet-enabled device to submit data to the cloud, which transforms sensor data into information that is useful for corporate decisions, machine-to-machine interactions, and educational research, resulting in increased global resource economization. It allows us to incorporate the power of the Internet of Things into our study in a simple and cost-effective manner. Its promising application platform will offer interactive, real-time data visualisation. The proposed work's functioning principle is depicted . The AD8232ECG sensor is attached to the ESP32 controller. All of the remaining components, as well as the electrodes, are arranged in an assembly box The voltage regulator is provided a 12v switched mode power supply (SMPS) because it will only provide the required voltage to the ESP32 controller.

The ESP32 controller will operate at a voltage of 3.3v. The ESP32 controller will run on a 3.3v power supply. The electrodes are put on the patient's chest, and the patient's heart beats are obtained in analogue form. As a result, the AD8232 ECG sensor is used to transform the data into digital form for efficient transmission. The collected digital data is then provided to the ESP32 controller, which responds on the signal based on embedded C language instructions. The signals are then connected to Ubidots, an IoT platform, which displays the ECG signal. It analyses the ECG data to determine any heart issues.. If the heartbeat rate exceeds a particular threshold (abnormal condition), the Ubidots platform sends an SMS message to the doctor or opens the Ubidots platform, alerting him to the patient's condition. Simultaneously, the buzzer and LED sound an alarm to alert the caretaker to the patient's status.

The ECG sensor utilised is a low-cost circuit that measures the heart's electrical activity. This electrical activity of the heart is recorded as an ECG (Electrocardiogram) and shown as an analogue readout. Because the ECG signal we obtained was noisy, we used an Op-amp, the AD8232, to quickly obtain a clean signal from the PR, QT, and ST intervals. The AD8232 is utilised to measure additional bio-potential measuring signals as well as signal conditioning for ECG. In the presence of noise, it is meant to magnify the bio-potential signal.

There are nine connecting pins and wires on the AD8232 Op-amp. Other connectors include LO+, LO-, OUTPUT, 3.3V, and GND, which are required to use the Op-Amp with an Arduino. This board also includes three lead electrodes: RA (Right Arm), LA (Left Arm), and RL (Right Leg) (Right Leg). The electrodes are put at a specific spot on the body to get an ECG signal since the required cardiac frequency can only be obtained at that location. The AD8232 ECG Sensor is connected to the ESP32 development kit. The AD8232 is powered by the ESP32 module, which provides 3.3V. The AD8232's output pin will be an analog signal. The VP pin of the ESP32 is then linked to this pin. Similarly, the AD8232's LO and LO+ are wired to ESP32's pins D3 and D2, respectively.

CHAPTER 4

COMPONENTS

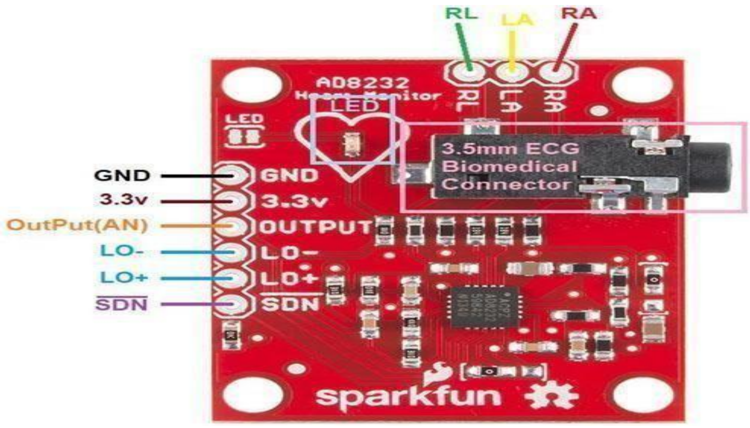
- ESP32 module.
- AD8232 sensor.
- Electrodes connected with 3.5mm jack.
- Breadboards.
- Connecting wires.

ESP32 MODULE:

The ESP32 is a versatile and widely used microcontroller module that belongs to the ESP (Espressif) family. It is designed for Internet of Things (IoT) applications and offers a combination of features that make it suitable for a variety of projects**.** ESP32 is a single 2.4 GHz Wi-Fi-and-Bluetooth combo chip designed with the TSMC low-power 40 nm technology. It is designed to achieve the best power and RF performance, showing robustness, versatility and reliability in a wide variety of applications and power scenarios.

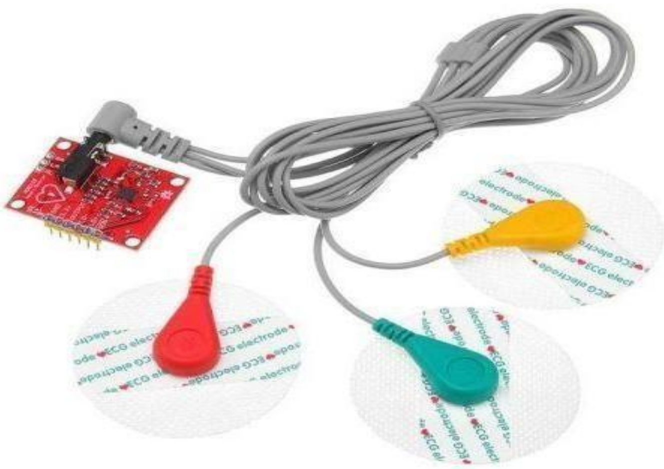
AD8232 SENSOR:

The AD8232 is an integrated signal conditioning block for ECG and other biopotential measurement applications. It is designed to extract, amplify, and filter small biopotential signals in the presence of noisy conditions, such as those created by motion or remote electrode placement.



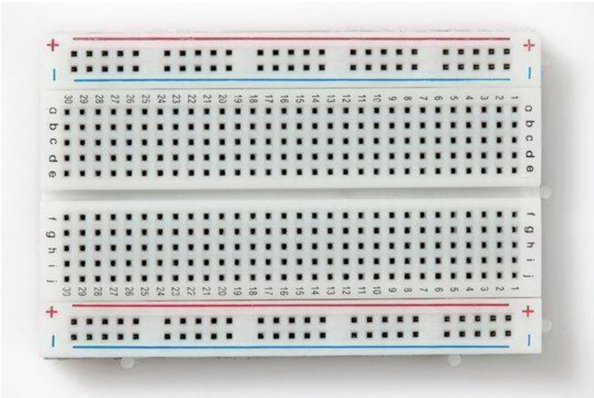
ELECTRODES:

An electrocardiogram (ECG) is one of the simplest and fastest tests used to evaluate the heart. Electrodes (small, plastic patches that stick to the skin) are placed at certain spots on the chest, arms, and legs. The electrodes are connected to an ECG machine by lead wires.



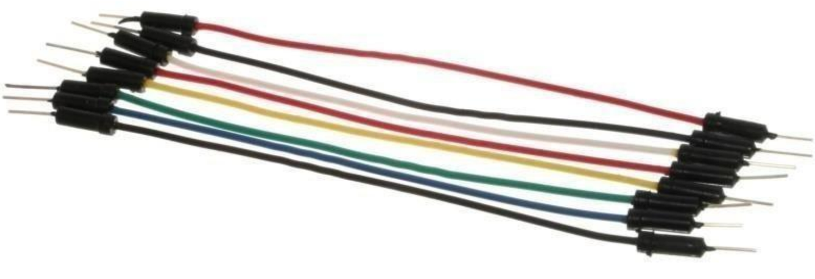
BREADBOARDS:

A breadboard (sometimes called a plugblock) is used for building temporary circuits. It is useful to designers because it allows components to be removed and replaced easily. It is useful to the person who wants to build a circuit to demonstrate its action, then to reuse the components in another circuit.



CONNECTING WIRES:

We use jumper wires with IoT boards, like Raspberry Pi 3, for connecting devices, so that we can write and execute our code which can be used to pass power and data using GPIO port. Jumper wires come in different categories and styles.



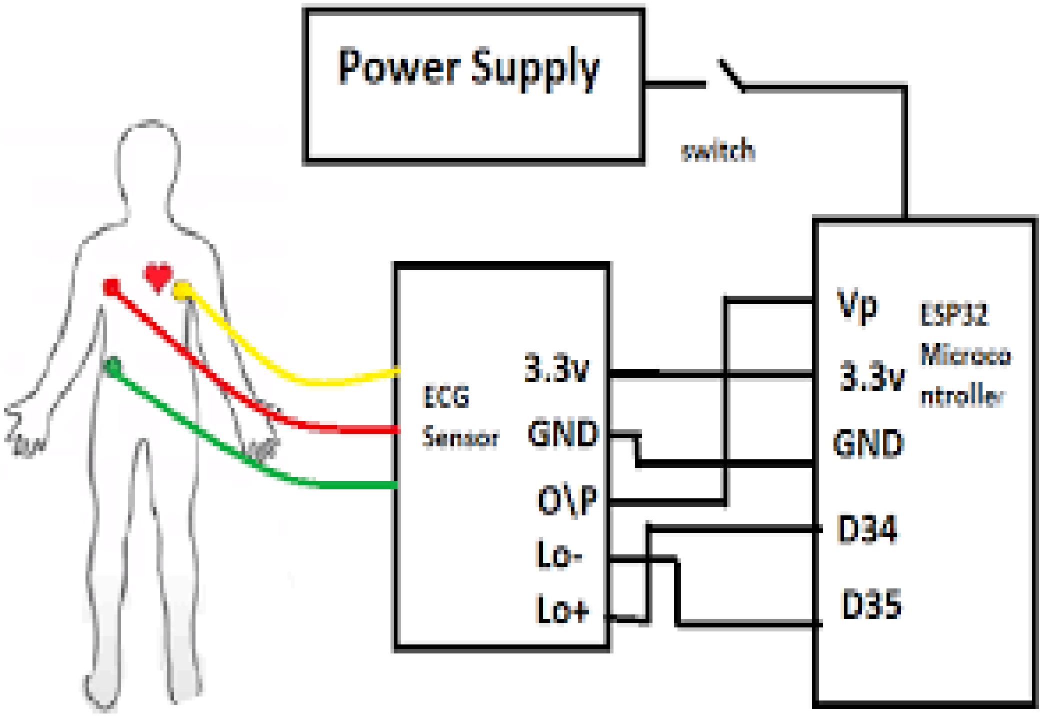
CHAPTER 5

SYSTEM IMPLEMENTATION

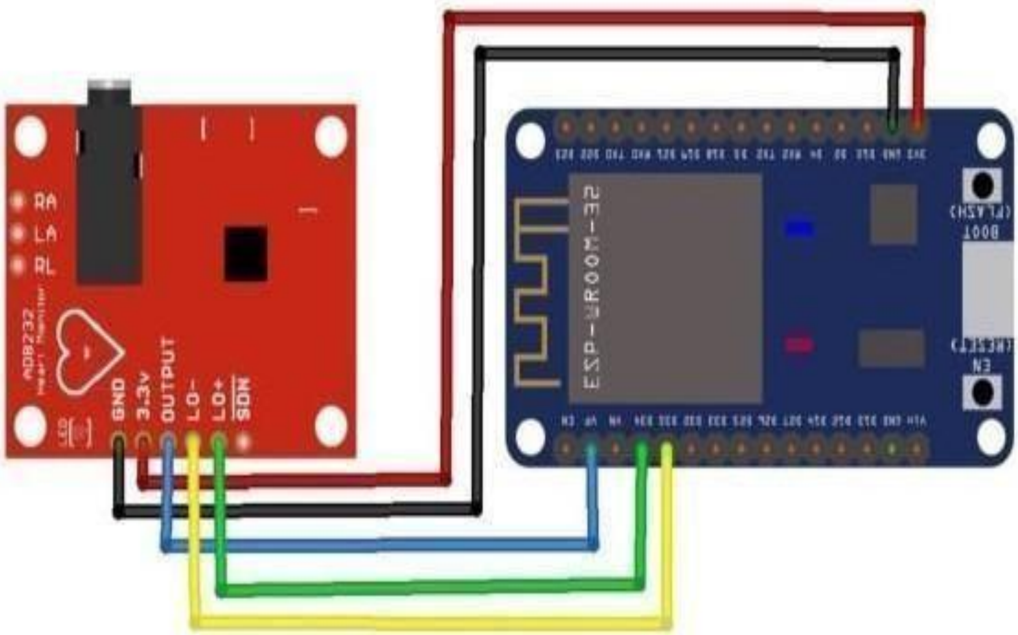
5.1 PIN CONFIGURATION:

The ESP32 chip features 34 physical GPIO pins (GPIO0 ~ GPIO19, GPIO21 ~ GPIO23, GPIO25 ~ GPIO27, and GPIO32 ~ GPIO39). Each pin can be used as a general-purpose I/O, or be connected to an internal peripheral signal.

5.2 BLOCK DIAGRAM:



5.3 CIRCUIT DIAGRAM:

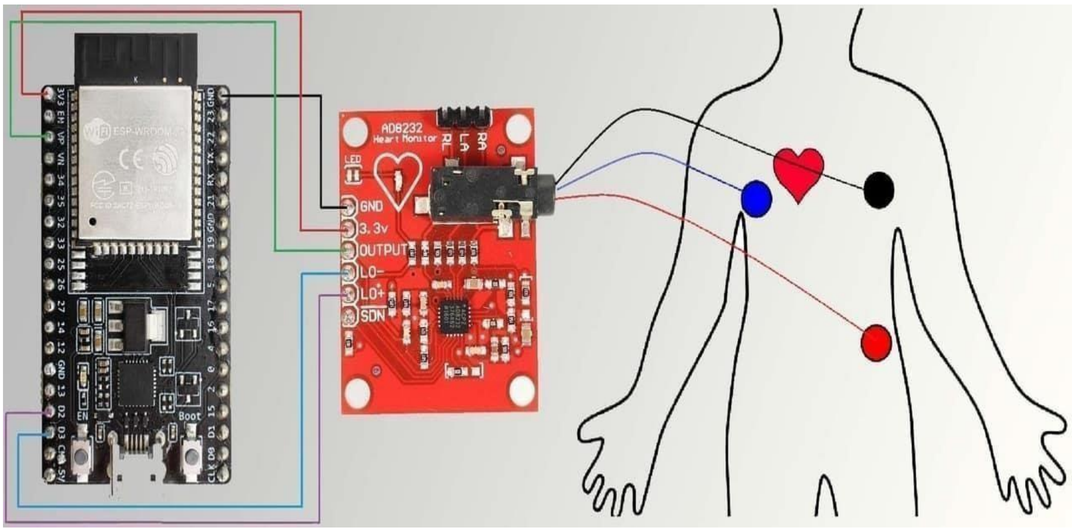


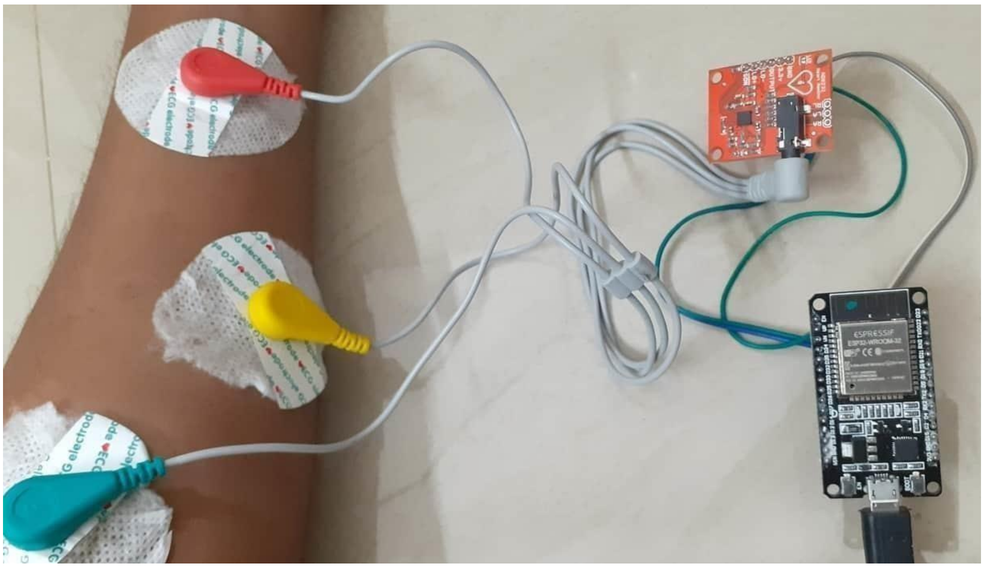
CHAPTER 6

SYSTEM WORKING

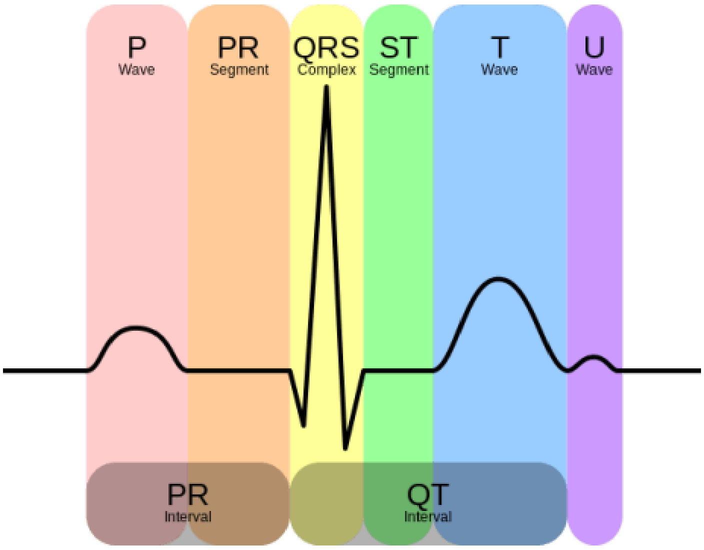
6.1 WORKING MODEL:

The integration of a human heartbeat rate monitoring system using a heart vibration sensor and IoT-based technologies are presented in this study. This sensor detects and records the human heartbeat. The ESP32 controller processes the read data before sending it to the Wi-Fi module for uploading to the Ubidots internet server platform for additional analysis and visualisation. When data is captured, it is processed and saved in real time, with a date and time tag. The input units in the proposed work are the ECG sensor unit, the power supply unit, and the user interface unit. The output unit is the Wi-Fi Module unit. The signals are monitored and controlled by the ESP32 controller unit. The integrated C programming language is used to programme the system. The heartbeat will be detected by the ECG sensor. The sensor's sensed data is sent to an analogue to digital converter (ADC), which converts it to a digital signal. After that, the generated digital signal is sent to the ESP32 controller. The ESP32 controller reacts to the signal based on instructions written in the embedded C programming language . The data is also delivered in real - time basis to the Wi-Fi module and then to the webserver (Ubidots) for more analytics and visualisation. To show the condition of the human heartbeat rate, the analysed data is updated continuously in real-time. On the input side, there's also a pulse sensor, an ECG sensor, and numerous manual buttons. The output is displayed on the ubidots dashboard. The Wi-Fi Module allows data to be uploaded to the cloud, and once there, the results may be accessed by logging into the server with a computer or smartphone.





Electrocardiogram(ECG):



The P wave of electrocardiogram represents atrial contraction. The PR interval is a measure of time from the onset of atrial contraction to the onset of ventricular contraction. The Q R S complex represents the complete depolarization of the ventricles, The S T segment represents the complete depolarization of the ventricles. elevation or depression of this segment may indicate heart muscle ischemia.The Q T interval represents the complete depolarization and repolarization of the ventricles.A prolonged interval is a risk factor for ventricular arrhythmias and sudden death.

6.2 SOFTWARE IMPLEMENTATION:

Integrating Arduino with an IoT project typically involves programming an Arduino board to read sensor data or control actuators and then communicating this information with an IoT platform or another device. Below are general steps to guide you through installing the Arduino IDE and integrating it with an IoT project:

Step 1: Install Arduino IDE Download Arduino IDE:

Go to the official Arduino website (https://www.arduino.cc/en/Main/Software) and download the latest version of the Arduino IDE suitable for your operating system (Windows, macOS, or Linux).

Install Arduino IDE:

Follow the installation instructions provided on the Arduino website to install the Arduino IDE on your computer.

Step 2: Set Up Arduino Board Connect ESP32 Board:

Connect your ESP32board to your computer using a USB cable.

Install Arduino Drivers (if needed):

Depending on your operating system, you might need to install drivers for the Arduino board. Check the official Arduino website or your board's documentation for instructions on driver installation.

Select Board and Port:

Open the Arduino IDE, and from the "Tools" menu, select the appropriate board model (e.g., Arduino Uno, ESP32) and the port to which your board is connected.

Step 3: Write and Upload Arduino Code Write Arduino Sketch:

In the Arduino IDE, create a new sketch (program) or open an existing one. Write the code to read sensor data or control actuators based on your IoT project requirements.

Verify and Upload Code:

Verify your code by clicking the checkmark icon in the Arduino IDE. If there are no errors, upload the code to your Arduino board by clicking the right arrow icon.

Step 4: Set Up IoT Platform Choose an IoT Platform:

Select an IoT platform to which your Arduino will send data. Common platforms include Blynk, ThingSpeak, AWS IoT, Google Cloud IoT, and many others.

Create an Account:

Sign up for an account on the chosen IoT platform.

Create a Project:

Create a new project on the IoT platform, and note down any authentication tokens or credentials provided by the platform.

Step 5: Integrate Arduino with IoT Platform Install IoT Library (if needed):

Depending on your IoT platform, you may need to install a library or package in the Arduino IDE that facilitates communication with the chosen platform.

Update Arduino Code:

Modify your Arduino code to include the necessary functions or libraries to connect to the IoT platform. This may involve setting up Wi-Fi or Ethernet connectivity and using specific IoT libraries.

Enter Credentials:

Enter the authentication tokens or credentials obtained from the IoT platform into your Arduino code.

Verify and Upload:

Verify and upload the modified code to your Arduino board.

Step 6: Monitor IoT Data Monitor Data on IoT Platform:

Open the dashboard or monitoring interface provided by the IoT platform to observe the data sent by your Arduino. This might include sensor readings, device status, or any other relevant information.

6.3 SOFTWARE CODING:

```
#include  <WiFi.h>

#include <WiFiUdp.h>

#include <PubSubClient.h>

#include <NTPClient.h>



#define WIFISSID "xxxxxxxxxxxxx" // Enter WifiSSID here

#define PASSWORD "xxxxxxx" // Enter password here

#define TOKEN "xxxxxxxxxxxxxxxxxxxxxxxxx" // Ubidots' TOKEN

#define MQTT_CLIENT_NAME "xxxxxxxxx" // MQTT client Name

// * Define Constants

#define VARIABLE_LABEL "xxxxxxxxxxx" // ubidots variable label

#define DEVICE_LABEL "xxxxxxxxxxxxxxxxxxxxx" // ubidots device label



#define SENSORPIN A0 // Set the A0 as SENSORPIN



char mqttBroker[] = "industrial.api.ubidots.com";

char payload[10000];
```

```
char topic[150];

// Space to store values to send

char str_sensor[10];

char str_millis[20];

double epochseconds = 0;

double epochmilliseconds = 0;

double current_millis = 0;

double current_millis_at_sensordata = 0;

double timestampp = 0;

int j = 0;

/****************************************

   Auxiliar Functions

 ****************************************/

WiFiClient ubidots;

PubSubClient client(ubidots);

WiFiUDP ntpUDP;

NTPClient timeClient(ntpUDP, "pool.ntp.org");



void callback(char* topic, byte* payload, unsigned int length) {
```

```
  char p[length + 1];

  memcpy(p, payload, length);

  p[length] = NULL;

  Serial.write(payload, length);

  Serial.println(topic);

}

void reconnect() {

  // Loop until we're reconnected

  while (!client.connected()) {

    Serial.println("Attempting MQTT connection...");



    // Attemp to connect

    if (client.connect(MQTT_CLIENT_NAME, TOKEN, "")) {

      Serial.println("Connected");

    } else {

      Serial.print("Failed, rc=");

      Serial.print(client.state());

      Serial.println(" try again in 2 seconds");
```

```
      // Wait 2 seconds before retrying
      delay(2000);
    }
  }
}

/****************************************
  Main Functions
****************************************/
void setup() {
  Serial.begin(115200);
  WiFi.begin(WIFISSID, PASSWORD);
  // Assign the pin as INPUT
  pinMode(SENSORPIN, INPUT);

  Serial.println();
  Serial.print("Waiting for WiFi...");

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

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

Serial.println("");
Serial.println("WiFi Connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
timeClient.begin();
client.setServer(mqttBroker, 1883);
client.setCallback(callback);
timeClient.update();
epochseconds = timeClient.getEpochTime();
epochmilliseconds = epochseconds * 1000;
Serial.print("epochmilliseconds=");
Serial.println(epochmilliseconds);
current_millis = millis();
Serial.print("current_millis=");
Serial.println(current_millis);
```

```
}

void loop() {
  if (!client.connected()) {
    reconnect();
    j = 0;
  }
  //sprintf(payload, "%s", "{\"ECG_Sensor_data\": [{\"value\":1234, \"timestamp\":
1595972075},{\"value\":1111, \"timestamp\": 1595971075},{\"value\":2222, \"timestamp\":
1595970075}]}");
  j = j + 1;
  Serial.print("j=");
  Serial.println(j);
  sprintf(topic, "%s%s", "/v1.6/devices/", DEVICE_LABEL);
  sprintf(payload, "%s", ""); // Cleans the payload
  sprintf(payload, "{\"%s\": [", VARIABLE_LABEL); // Adds the variable label
  for (int i = 1; i <= 3; i++)
  {
    float sensor = analogRead(SENSORPIN);
    dtostrf(sensor, 4, 2, str_sensor);
```

```
  sprintf(payload, "%s{\"value\":", payload); // Adds the value

  sprintf(payload, "%s %s,", payload, str_sensor); // Adds the value

  current_millis_at_sensordata = millis();

  timestampp = epochmilliseconds + (current_millis_at_sensordata - current_millis);

  dtostrf(timestampp, 10, 0, str_millis);

  sprintf(payload, "%s \"timestamp\": %s},", payload, str_millis); // Adds the value

  delay(150);

}

float sensor = analogRead(SENSORPIN);

dtostrf(sensor, 4, 2, str_sensor);

current_millis_at_sensordata = millis();

timestampp = epochmilliseconds + (current_millis_at_sensordata - current_millis);

dtostrf(timestampp, 10, 0, str_millis);

sprintf(payload, "%s{\"value\":%s, \"timestamp\": %s}]}", payload, str_sensor, str_millis);

Serial.println("Publishing data to Ubidots Cloud");

client.publish(topic, payload);

Serial.println(payload);

// client.loop();
```

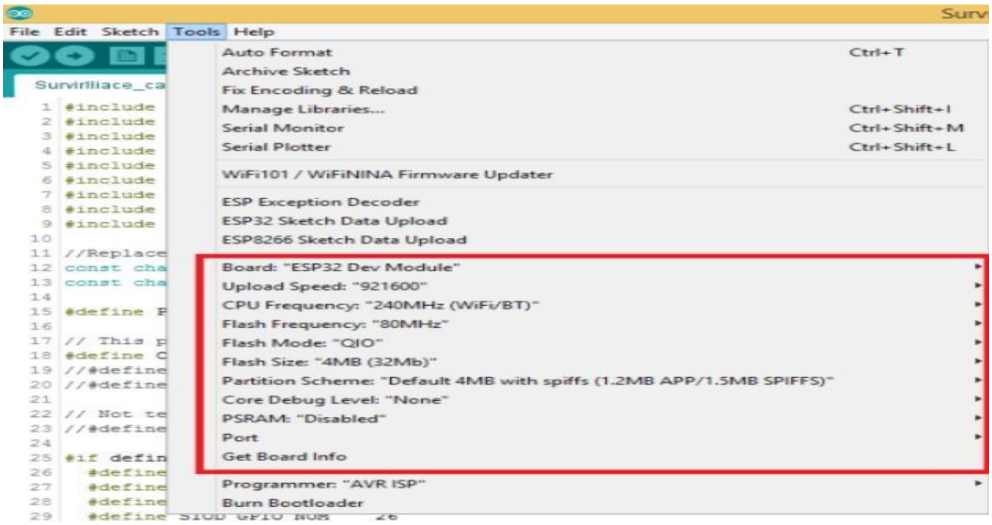
}

EXPLANATION:

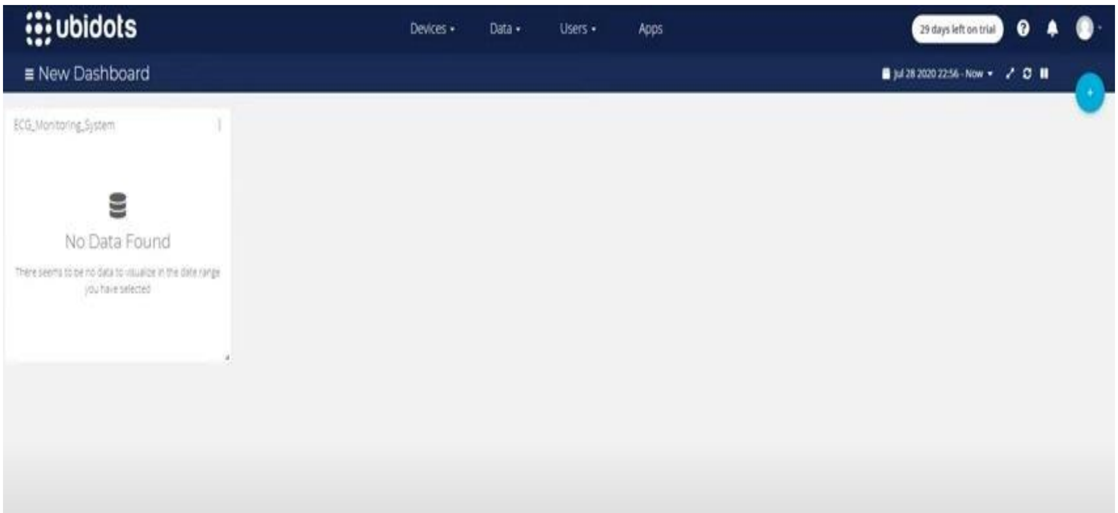
#define WIFISSID "xxxxxxxxxxxxxxxxx" // WifiSSID Should be entered

#define PASSWORD "xxxxxxxxxx" // password Should be entered

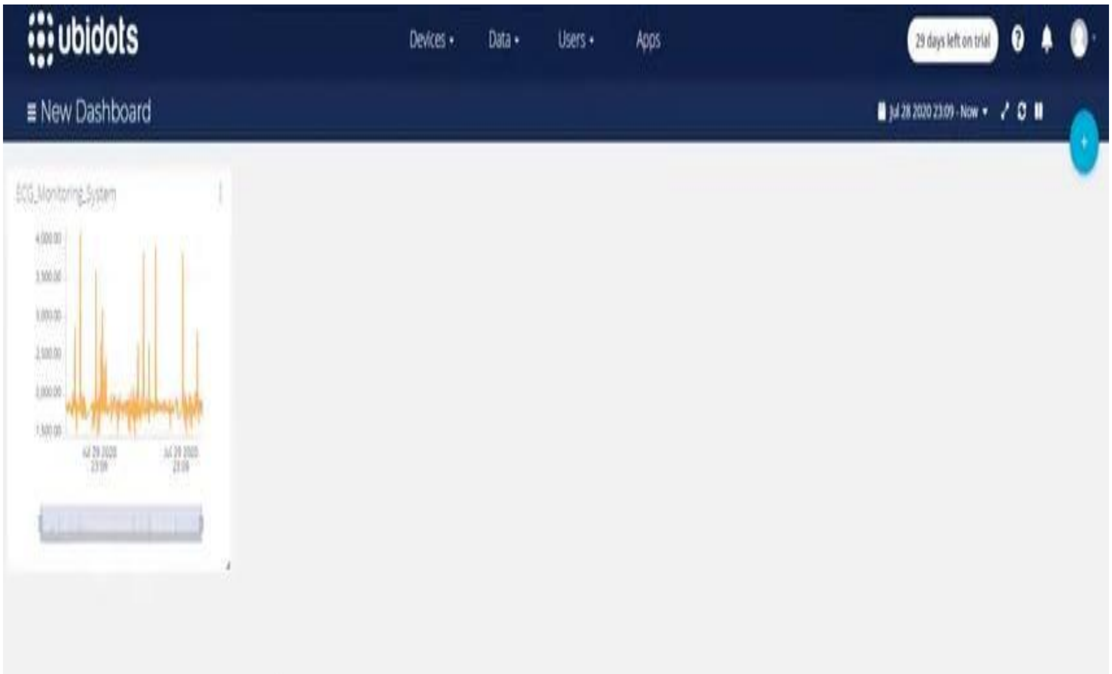
Connect ESP32 module with Laptop/computer. Check the Uploading configurations like board: ESP32 Dev Module, Uploading Speed, and select the right port. Then upload the code.

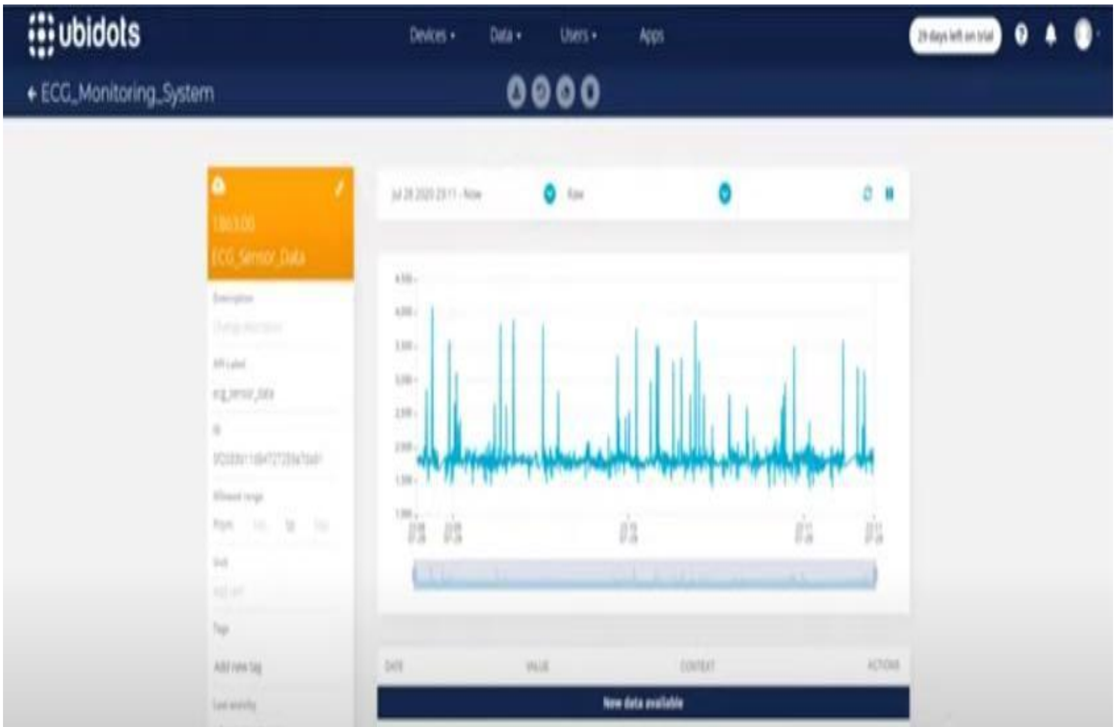


Before uploading and connecting the code the following result will be obtained



After proper uploading the following result will be obtained.





CHAPTER 7

ADVANTAGES

Remote Monitoring:

Patient Convenience: IoT-enabled ECG devices allow patients to monitor their heart health from the comfort of their homes. This is particularly beneficial for individuals with chronic conditions or those who live in remote areas, reducing the need for frequent hospital visits.

Continuous Monitoring: Continuous monitoring provides a more comprehensive view of the patient's heart activity, allowing healthcare professionals to detect irregularities or trends over time.

Timely Detection and Intervention:

Real-time Alerts: IoT devices can provide real-time alerts to healthcare providers or patients when abnormal heart patterns are detected. This enables prompt intervention, potentially preventing serious health events.

Early Diagnosis: Continuous monitoring facilitates the early detection of cardiac issues, allowing for timely diagnosis and treatment, which can significantly improve patient outcomes.

Data Accuracy and Precision:

High-Quality Data: IoT devices can capture high-quality ECG data, minimizing artifacts and inaccuracies often associated with traditional monitoring methods.

Data Analytics: The collected data can be analyzed using advanced analytics, providing insights into trends, patterns, and potential risk factors. This data-driven approach enhances the accuracy of diagnoses and treatment plans.

Cost Efficiency:

Reduced Healthcare Costs: By enabling remote monitoring and early intervention, IoT-based ECG monitoring can potentially reduce healthcare costs associated with emergency room visits and hospital stays. It also helps in preventing complications that may require expensive treatments.

Patient Empowerment:

Self-Management: Patients actively participating in their healthcare by monitoring their ECG at home can become more engaged in managing their conditions. This empowerment may lead to better adherence to treatment plans and lifestyle modifications.

Accessibility and Scalability:

Widespread Access: IoT-enabled ECG devices can be deployed on a large scale, providing access to cardiac monitoring in diverse settings, from homes to healthcare facilities. This increased accessibility is crucial for managing population health.

Wireless Connectivity:

The ESP32 comes with built-in Wi-Fi and Bluetooth capabilities. This enables wireless communication, allowing ECG data to be transmitted in real-time to other devices or a centralized monitoring system.

Compact Size and Portability:

The ESP32 is a compact and lightweight microcontroller, making it suitable for portable and wearable ECG monitoring devices. This is particularly advantageous for patients who need continuous monitoring without being tethered to a fixed location.

Low Power Consumption:

The ESP32 is designed with power efficiency in mind. This is crucial for battery-powered devices, ensuring that ECG monitoring can be carried out for extended periods without frequent battery replacements.

Cost-Effectiveness:

The ESP32 is known for its affordability, making it a cost-effective solution for ECG monitoring devices. This can be particularly important for widespread deployment in healthcare systems or for individual use.

CHAPTER 8

APPLICATIONS

Wearable Health Devices:

Integration of the ESP32 into wearable devices, such as smartwatches or fitness trackers, enables continuous ECG monitoring for individuals. This is particularly useful for those interested in tracking their heart health during daily activities.

Remote Patient Monitoring:

ESP32-based ECG monitors can be employed for remote patient monitoring, allowing healthcare providers to track patients' heart health in real-time. This is beneficial for individuals with chronic conditions or those recovering from cardiac-related procedures.

Home Health Monitoring Systems:

ESP32 can be part of home health monitoring systems, allowing users to perform ECG measurements at home. The data can be transmitted to healthcare providers for analysis, promoting proactive healthcare management.

Fitness and Sports Applications:

Athletes and fitness enthusiasts can benefit from ESP32-based ECG monitoring during training sessions. This data can be used to analyze heart rate variability, optimize workout routines, and prevent overtraining.

Emergency Response Systems:

ESP32-connected ECG monitors can be integrated into emergency response systems. In case of a cardiac emergency, the device can automatically send ECG data to emergency services or designated contacts, facilitating rapid response.

Smart Healthcare Infrastructure:

Integration of ESP32 in healthcare infrastructure, such as smart hospital beds or smart ambulances, allows for continuous ECG monitoring during patient transport or hospital stays.

Occupational Health Monitoring:

In industries where workers are exposed to physically demanding or stressful conditions, ESP32-based ECG monitoring can help assess and manage the impact of work-related stress on employees' cardiovascular health.

Educational Tools:

ESP32-based ECG monitoring devices can be used as educational tools in medical and healthcare training. They provide a hands-on experience for students to understand ECG interpretation and the use of IoT in healthcare.

Research and Prototyping:

Researchers and developers can use the ESP32 for prototyping and developing ECG monitoring solutions. Its versatility and programmability make it a suitable platform for exploring new features and functionalities.

Low-Cost Healthcare Solutions:

ESP32's affordability makes it suitable for creating low-cost ECG monitoring solutions, addressing cost constraints in healthcare systems and promoting accessibility to cardiovascular monitoring.

Data Logging and Analysis:

ESP32 can be utilized for continuous data logging of ECG signals. The collected data can be analyzed locally or transmitted to a centralized system for further analysis and reporting.

Personal Health Records (PHR):

Integrating ESP32-based ECG monitoring into personal health record systems allows individuals to maintain a comprehensive record of their cardiovascular health over time.

CHAPTER 9

FEATURE ENHANCEMENT

Continued advancements in ECG monitoring using IoT can lead to more sophisticated and user-friendly solutions. Here are some feature enhancements that could be implemented:

Real-Time Cloud Connectivity:

Enable seamless real-time connectivity to cloud platforms. This allows for immediate data storage, analysis, and access from anywhere, enhancing remote monitoring capabilities.

Edge Computing for On-Device Processing:

Implement edge computing capabilities to perform advanced signal processing directly on the device. This reduces reliance on constant cloud connectivity and allows for quicker response times.

Machine Learning for Anomaly Detection:

Integrate machine learning algorithms to analyze ECG data patterns. This can improve anomaly detection accuracy and enable early identification of potential cardiac issues.

Personalized Health Insights:

Provide personalized insights based on individual ECG data. This could include recommendations for lifestyle modifications, exercise routines, or stress management techniques tailored to the user's specific health profile.

Integration with Wearable Devices:

Enhance compatibility with various wearable devices, such as smartwatches or fitness trackers, for a more seamless user experience. Users can consolidate health data from different sources in a single platform.

Biometric Authentication for Data Security:

Implement biometric authentication methods (such as fingerprint or facial recognition) to ensure secure access to ECG data, maintaining user privacy and complying with healthcare data protection standards.

Smart Notifications and Alerts:

Improve notification systems to provide intelligent alerts. These alerts can be based on user-specific thresholds, abnormal ECG patterns, or reminders for regular check-ups.

Long-Term Trend Analysis:

Incorporate features for long-term trend analysis of ECG data. This allows users and healthcare professionals to identify trends, monitor changes over time, and make informed decisions about treatment or lifestyle adjustments.

Gamification for User Engagement:

Introduce gamification elements to encourage regular ECG monitoring. Users could earn rewards or participate in challenges, fostering engagement and adherence to monitoring routines.

Integration with Telemedicine Platforms:

Facilitate integration with telemedicine platforms, allowing users to share ECG data with healthcare providers during virtual consultations. This enhances communication and enables more informed remote healthcare.

Multimodal Health Monitoring:

Expand monitoring capabilities to include other health parameters, such as blood pressure, oxygen saturation, or sleep patterns. This provides a more holistic view of overall health.

Language Localization and Accessibility Features:

Support multiple languages and incorporate accessibility features for users with diverse needs. This ensures inclusivity and broadens the reach of ECG monitoring solutions.

Automatic Calibration and Electrode Check:

Implement features for automatic calibration and electrode check to ensure the accuracy of ECG measurements. This reduces the risk of inaccurate readings due to electrode issues.

Smart Data Visualization:

Enhance data visualization tools to provide users and healthcare professionals with clear and intuitive representations of ECG data trends, making it easier to interpret and analyze.

Continuous Firmware Updates:

Allow for over-the-air (OTA) firmware updates to ensure that devices can receive the latest features, security patches, and improvements without requiring physical intervention.

CHAPTER 10

CONCLUSION

Only with AD8232 ECG Sensor and the ESP32 development kit, we suggested an IoT-based heartbeat monitoring system. Heart ailments have become more prevalent in recent decades, and many individuals have died as a result of these illnesses. As a result, heart illness should not be regarded lightly. Heart disease could be avoided by studying and constantly monitoring the ECG signal early on. The observed ECG signal is examined and analysed.

Early detection, prediction, and management of cardiac disease are all aided by IoT-based statistical frameworks for heart monitoring. This study investigates an IoT-connected lowpower wireless sensor interaction technology for long-term cardiac parameter monitoring. Regular usage of the gadget is extremely beneficial for early diagnosis of heart ailments as well as lowering the severity of damage and mortality rates associated with cardiovascular diseases. Additional health monitoring systems, such as temperature measurement, Pulse Rate, Diabetes, and so on, can be constructed utilising IoT in a similar way to this userfriendly ECG surveillance system, and will significantly help to reduce serious health complications to a certain extent. ECG monitoring systems are constantly being implemented using new technologies including as deep learning, AI, Big Data, and IoT to deliver a cost- effective, completely linked, and sophisticated monitoring system. Empowering technologies provide up a wide variety of possibilities for ECG monitoring system improvement.IoT introduces distant, unconstrained communication and services that make use of data to enable fast, meaningful, and crucial lifestyle decisions. Furthermore, fog computing and cloud processing contribute to greater efficiency and the fulfilment of multiple in-demand extensible application services. Furthermore, blockchain enables security for numerous transactions across the many elements of the ECG monitoring system's development in a distributed environment.

CHAPTER 11

REFERENCES

[1.] Mohamed Adel Serhani Hadeel T. El Kassabi Heba Ismail and Alramzana Nujum Navaz ECG Monitoring Systems: Review, Architecture, Processes, and Key Challenges. Sensors 2020, 20, 1796; doi:10.3390/s20061796 (2020)

[2.] Scutti, S. Nearly Half of US Adults Have Cardiovascular Disease, Study Says. CNN, Cable News Network.(2019)

[3.] European Cardiovascular Disease Statistics 2017; The European Heart Network (EHN): Brussels, Belgium, 2017; Available online: http://www.ehnheart.org/cvdstatistics.html (2020)

[4.] L. Wang, G.-Z. Yang, J. Huang, J. Zhang, L. Yu, Z. Nie et al., “A wireless biomedical signal interface system-on-chip for body sensor networks,” IEEE Trans. Biomed. Circuits Syst., vol. 4, no. 2, pp. 112–117, (2010)

[5.] Abadi M, Subramanian R, Kia S et al. (2015) DECAF: MEG-based multimodal database for decoding affective physiological responses.IEEE Transactions on Affective Computing 6:209-222(2015)

[6.] Yakut, Onder, Serdar Solak, and Emine Dogru Bolat. "Implementation of a Web-Based Wireless ECG Measuring and Recording System."In 17th International Conference on Medical Physics and MedicalSciences, Istanbul, vol. 9, no. 10, pp. 815-818.(2015)

[7.] H. T. Yew, Y. Aditya, H. Satrial, E. Supriyan and Y.W Hau,“Telecardiology system for fourth generation heterogeneous wireless networks,” ARPN J. Eng. Appl. Sci., vol. 10, no. 2,(2015)

[8.] Azariadi, Dimitra, Vasileios Tsoutsouras, Sotirios Xydis, and Dimitrios Soudris. "ECG signal analysis and arrhythmia detection on IoT wearable medical devices." In Modern Circuits and Systems Technologies (MOCAST), 2016 5th International Conference on, pp. 1-
4. IEEE, 2016

[9.] H. T. Yew, E. Supriyanto, M. H. Satria, and Y. W. Hau, “Adaptive network selection mechanism for telecardiology system in developing countries,” in 3rd IEEE EMBS International Conference on Biomedical and Health Informatics, BHI (2016)

[10.] Palattella, M., Dohler, M., Grieco, A., and et al., Internet of Thingsthe 5G era: Enablers, architecture, and business models. IEEE J. Select. Areas Commun. 34:510–527,(2016).

[11.] Wang Y et al. Design and evaluation of a novel wireless reconstructed 3-lead ECG monitoring system. In: Proc. IEEE 2013 Biomedical Circuits and Systems Conference (BioCAS), Rotterdam,(2013).

[12.] Parmveer Singh Ashish Jasuja , School of VLSI Design and Embedded System NIT Kurukshetra International Conference on Computing, Communication and Automation (ICCCA2017).

[13.] T. Yokotani and Y. Sasaki, “Transfer protocols of tiny data blocks in IoT and their performance evaluation,” 2016 IEEE 3rd World Forum Internet Things, WF-IoT 2016, pp. 54–57, (2017).

[14.] International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering Vol. 6, Issue 2, (2017)

[15.] Tamanna Shaown, Imam Hasan, Md.Muradur Rahman Mim 1st International Conference on Advances in Science, Engineering and Robotics Technology 2019 (ICASERT 2019)

[16.] Zhe Yang Senior Member, IEEE An IoT-cloud Based Wearable ECG Monitoring System for Smart Healthcare. IEEE Publications. (2019)

[17.] Hoe Tung Yew University Malaysia Sabah 2020 16th IEEE International Colloquium on Signal Processing & its Applications (CSPA 2020), 28-29 (Feb. 2020)

[18.] (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 12, No. 6, 2021-Internet of Things (IoT) based ECG System for Rural Health Care by Md. Obaidur Rahman1 Department of CSE, DUET, EUB, Gabtoli, Dhaka, Bangladesh(2021)

[19.]Jovanov, E.; Milenkovic, A. Body area networks for ubiquitous healthcare applications: Opportunities and challenges. J. Med. Syst. 2011, 35, 1245–1254(2011)