MANTECH Publications

Journal of Internet of Things and Information Technology
Volume 8, Issue 1, January-April, 2025

Smart Health Care System

Ravindra Dhongadi1, Tanuja Kamble[2], Nimisha Patil[2], Anushka Hajare[2]

Professor[1], Students[2]

Department of Electronics and Telecommunication Engineering

Sanjay Ghodawat Institute, Atigre, Maharashtra, India

Correspondence Author Email Id: *nimishapatil420@gmail.com*

Abstract

Cardiovascular diseases (CVDs) are a leading cause of mortality worldwide, necessitating early detection and continuous monitoring for effective prevention and treatment. The Smart Heart Care System is an advanced, AI-driven healthcare solution designed to monitor, analyze, and predict heart health conditions in real time. This system integrates wearable sensors, Internet of Things (IoT) technology, and machine learning algorithms to track vital parameters such as heart rate, blood pressure, ECG, and oxygen levels. The collected data is processed and analyzed to detect anomalies, predict potential risks, and provide instant alerts to patients and healthcare professionals.

Additionally, the system employs cloud-based storage for secure and remote access to patient records, enabling telemedicine and timely medical interventions. With AI-driven predictive analytics, the Smart Heart Care System enhances early diagnosis and personalized healthcare, reducing hospital visits and improving patient outcomes. This innovative approach to heart health management has the potential to revolutionize cardiovascular care, making it more accessible, efficient, and proactive.

Keywords: *Cardiovascular diseases, AI-driven healthcare, Smart Heart Care System, real-time monitoring, wearable sensors, Internet of Things (IoT), machine learning, heart rate monitoring, ECG analysis, predictive analytics, telemedicine, cloud-based healthcare, anomaly detection, personalized healthcare, remote patient monitoring.*

Page 1-15 © MANTECH PUBLICATIONS 2025. All Rights Reserved

INTRODUCTION

Advancements in healthcare technology have significantly improved the diagnosis, monitoring, and management of various health conditions. Among these, cardiovascular diseases (CVDs) remain a leading cause of morbidity and mortality worldwide, requiring continuous monitoring and early intervention to prevent severe complications. Traditional healthcare systems often rely on periodic check-ups, which may not provide timely insights into a patient's health status.

The Smart Heart Care System is an innovative, AI-driven solution that leverages modern technologies such as the Internet of Things (IoT), wearable sensors, cloud computing, and machine learning to offer real-time monitoring and predictive analysis of heart health. This system continuously tracks vital parameters like heart rate, blood pressure, electrocardiogram (ECG), and oxygen levels, enabling proactive detection of irregularities and early warnings for potential cardiac events.

By integrating cloud-based storage and telemedicine capabilities, the system ensures seamless access to medical records for healthcare professionals, allowing timely interventions and remote consultations. The incorporation of AI-powered analytics further enhances the system's ability to provide personalized healthcare, reducing hospital visits and improving patient outcomes.

This paper explores the design, functionality, and impact of the Smart Heart Care System, highlighting its potential to revolutionize cardiovascular disease management through intelligent, real-time health monitoring.

LITERATURE SURVEY

The rapid advancement of healthcare technology has led to the development of smart health monitoring systems that leverage artificial intelligence (AI), the Internet of Things (IoT), and wearable devices for real-time health tracking. Several studies and research papers have explored various aspects of smart healthcare systems, particularly for cardiovascular disease (CVD) monitoring and early diagnosis.

Page 1-15 © MANTECH PUBLICATIONS 2025. All Rights Reserved

IoT-Based Health Monitoring Systems

IoT-based healthcare solutions have gained significant attention due to their ability to provide continuous health monitoring. Research by Kumar et al. (2021) discusses how IoT-enabled wearable sensors collect physiological parameters such as heart rate, blood pressure, and ECG in real time. The data is then transmitted to cloud platforms for further analysis and storage. This approach enhances remote patient monitoring and enables timely medical intervention.

AI and Machine Learning in Smart Healthcare

The role of AI and machine learning (ML) in predicting and diagnosing cardiovascular diseases has been explored in various studies. A study by Patel et al. (2020) highlights the effectiveness of ML algorithms in analyzing ECG signals and detecting abnormalities such as arrhythmias. AI- powered predictive analytics can assess the risk of heart attacks based on historical patient data, improving preventive care.

Wearable Sensor Technologies for Heart Health Monitoring

Wearable health devices such as smart watches and biosensors have revolutionized real-time health tracking. According to a study by Zhang et al. (2019), wearable devices equipped with ECG and PPG sensors can detect irregular heart rhythms and provide alerts to patients and doctors. The integration of Bluetooth and cloud storage enables continuous data transmission for remote monitoring.

Cloud-Based Healthcare and Telemedicine

Cloud computing plays a crucial role in smart healthcare by providing secure storage and easy access to patient health records. Research by Sharma et al. (2022) discusses how cloud-based healthcare systems facilitate telemedicine, allowing doctors to diagnose and prescribe treatments remotely. This approach is particularly beneficial for patients in rural or remote areas with limited access to healthcare facilities.

Challenges and Future Directions in Smart Healthcare

Despite the numerous advantages, smart healthcare systems face challenges such as data security, interoperability, and the need for high-speed internet connectivity. A study by Ramesh et al. (2023) emphasizes the importance of cybersecurity measures to protect patient data and the need for standardized protocols to enhance system integration and efficiency.

Page 1-15 © MANTECH PUBLICATIONS 2025. All Rights Reserved

PROPOSED SYSTEM

The Smart Care System (SCS) is an intelligent healthcare solution that uses smart sensors, wearable devices, AI-driven analytics, and cloud-based storage to monitor patient health, provide alerts, and assist in decision-making for medical professionals and caregivers. The Smart Heart Care System is designed to provide real-time monitoring, early diagnosis, and predictive analysis of cardiovascular health using advanced technologies such as Artificial Intelligence (AI), the Internet of Things (IoT), wearable sensors, and cloud computing. The proposed system aims to overcome the limitations of traditional healthcare by enabling continuous health tracking and remote medical intervention.

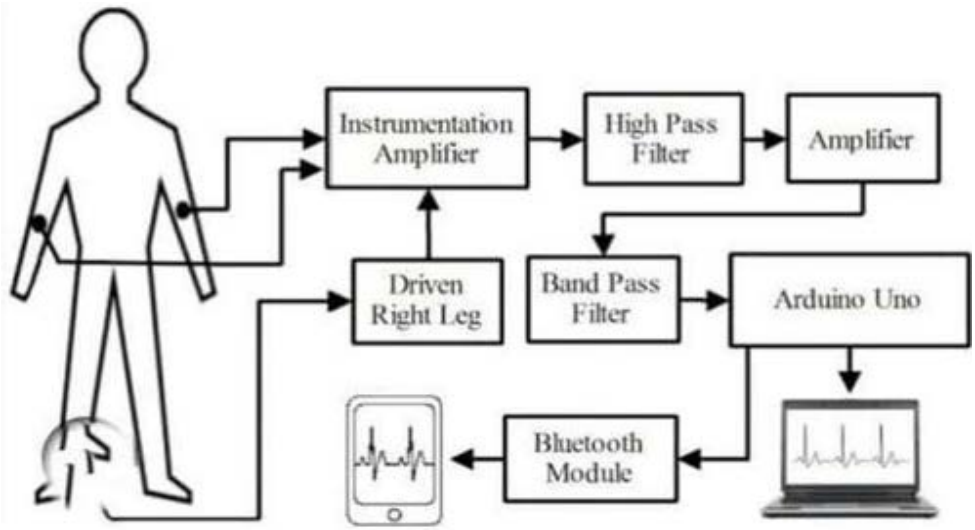


Figure no.: 1

Node MCU

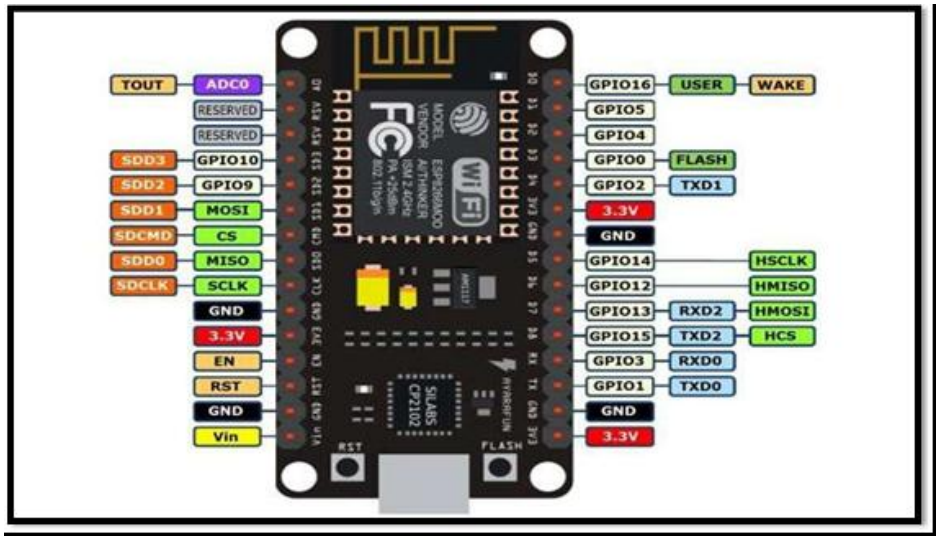


Figure no.: 2

Page 1-15 © MANTECH PUBLICATIONS 2025. All Rights Reserved

Node MCU is an open-source platform for Internet of Things (IoT) development, featuring both firmware (based on the ESP8266 Wi-Fi SoC) and hardware (based on the ESP-12 module), allowing users to program with Lua scripting language or Arduino IDE.

Key Features and Information

- **Firmware:** NodeMCU firmware is open-source and runs on the ESP8266 Wi-Fi SoC.
- **Hardware:** The hardware is based on the ESP-12 module, a popular and versatile Wi-Fi module.
- **Programming:** NodeMCU can be programmed using Lua scripting language or Arduino IDE.
- **Open Source:** Both the firmware and hardware designs are open-source, allowing for customization and community contributions.
- **IoT Applications:** NodeMCU is well-suited for IoT projects, enabling easy prototyping and development of wireless applications.
- **ESP8266:** The core of NodeMCU is the ESP8266, a low-cost Wi-Fi chip with built-in TCP/IP networking software and microcontroller capability.
- **Versatility:** NodeMCU is a versatile platform that can be used for a wide range of IoT applications, from simple sensor projects to more complex networked devices.
- **Arduino IDE Compatibility:** NodeMCU can be programmed using the Arduino IDE, making it easier for users familiar with the Arduino platform to get started.
- **Lua Scripting:** NodeMCU uses Lua, a lightweight and embeddable scripting language, which is easy to learn and use.
- **Open-Source Projects:** NodeMCU utilizes many open-source projects, such as lua-cjson and SPIFFS, a flash file system for embedded controllers.
- **ESP32 Support:** Support for the 32-bit ESP32 has also been implemented.
- **Development Kits:** NodeMCU is available as a development kit, which includes the hardware and firmware, making it easy to get started with IoT projects.
- **Pinout:** The NodeMCU board has various pins for different functionalities, including GPIO (General Purpose Input/Output) pins, ADC (Analog to Digital Converter) pin, SPI (Serial Peripheral Interface) pins, I2C (Inter-Integrated Circuit) pins, and UART (Universal Asynchronous Receiver Transmitter) pins.

Page 1-15 © MANTECH PUBLICATIONS 2025. All Rights Reserved

- **Power Requirements:** The NodeMCU board typically operates on a 3.3V voltage level and can be powered via a micro USB port.

ELECTRODE



Figure no.: 3

Electrode" has multiple meanings, including a small device that conducts electricity and a platform for building applications. As a conductor: An electrode is a small device that conducts electricity and can carry current into non-metals and poor conductors. They are used in many applications, including medical devices, fuel cells, and electrochemical cells. Electrodes can be positively charged (anode) or negatively charged (cathode).

AD8232 ECG SENSOR

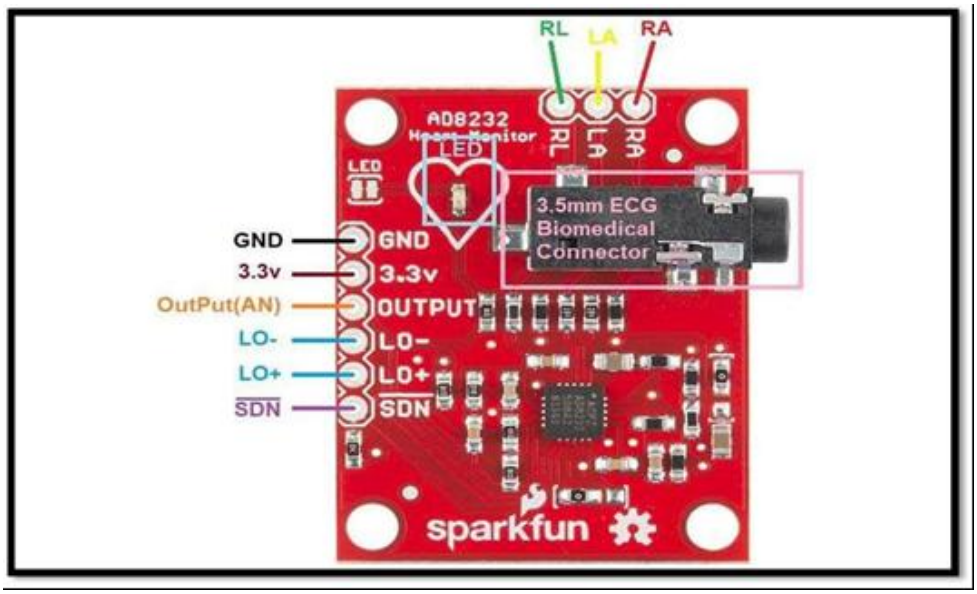


Figure no.: 4

Page 1-15 © MANTECH PUBLICATIONS 2025. All Rights Reserved

The AD8232 ECG sensor is a signal conditioning block that amplifies, filters, and extracts small electrical signals from the heart. It's used to create an electrocardiogram (ECG) that shows the heart's electrical activity.

How it works?

- Electrodes are placed on the chest, arms, or legs.
- The electrodes pick up electrical signals from the heart.
- The AD8232 amplifies, filters, and extracts the signals.
- The AD8232 creates an analog output.
- The output is connected to a microcontroller or analog-to-digital converter (ADC).
- The microcontroller or ADC samples the signal and sends it to a computer for display.

Features

- The AD8232 is designed to reduce noise from motion or electrode placement.
- It can implement a high-pass filter to remove motion artifacts.
- It can implement a low-pass filter to remove additional noise.
- It can be used with an Arduino, ESP32, or other microcontroller.

Applications

- The AD8232 can be used to detect heart attacks, abnormal heart rhythms, and cholesterol buildup.

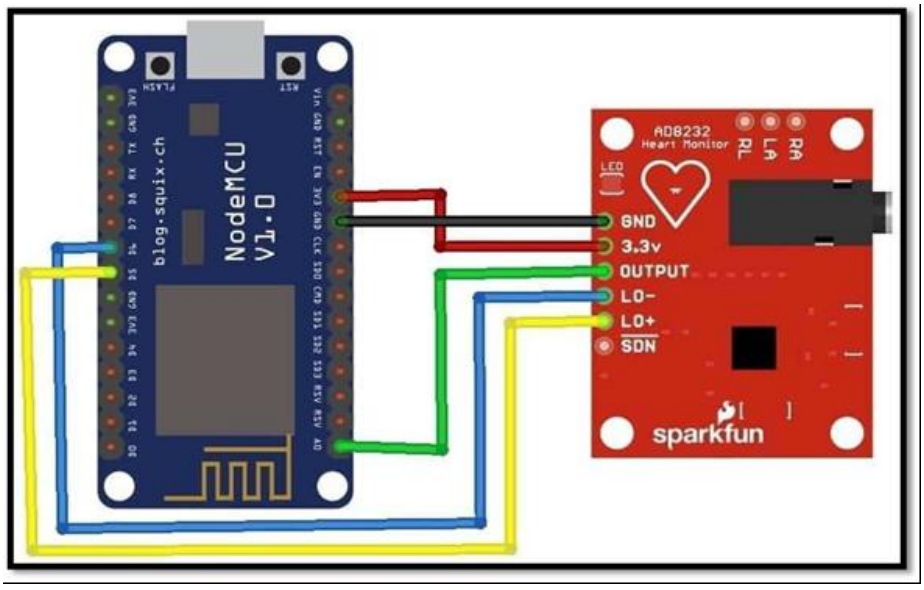


Figure no.: 4

Page 1-15 © MANTECH PUBLICATIONS 2025. All Rights Reserved

MAX30102 Sensor

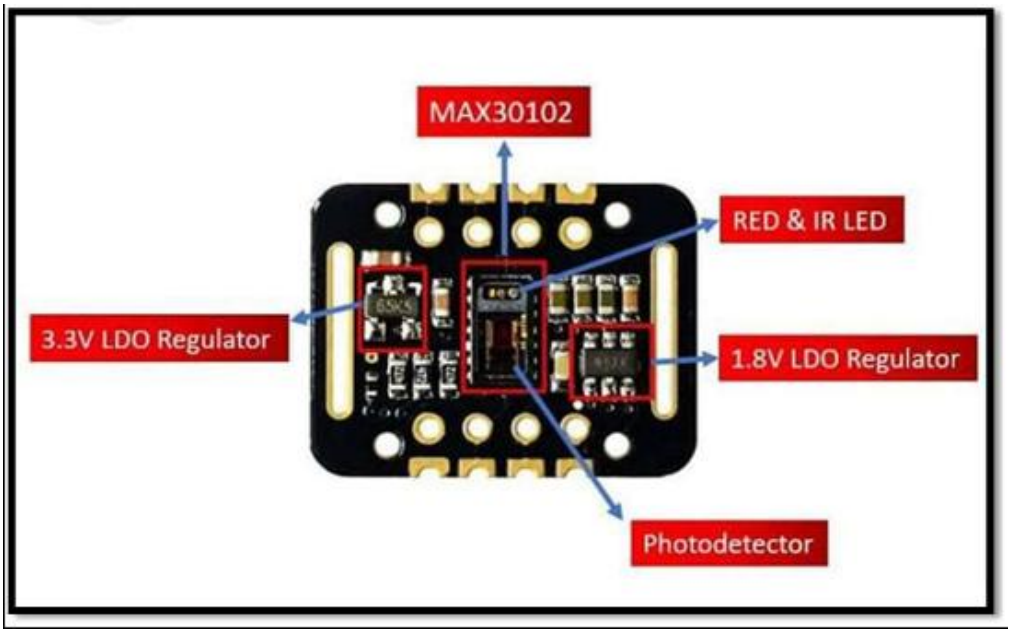


Figure no: 5

The MAX30102 sensor, a pulse oximeter and heart rate monitor, works by shining red and infrared light through the skin and measuring the amount of light reflected back, a process called photoplethysmography (PPG). This allows it to detect heart rate and blood oxygen levels.

How it Works?

- The sensor uses two LEDs: a red LED (660 nm) and an infrared LED (880 nm).
- These LEDs shine light through the skin, and a photo detector measures the amount of light that is reflected back.
- The amount of light absorbed or reflected varies depending on the amount of oxygenated and deoxygenated blood in the tissues.
- This variation in light absorption creates a waveform that can be used to determine heart rate and blood oxygen saturation (SpO2).

HEART RATE MEASUREMENT

- As blood is pumped through the arteries with each heartbeat, the amount of reflected light changes, creating a pulsating waveform at the output of the photo detector.
- By analyzing this waveform, the heart rate can be determined.

Page 1-15 © MANTECH PUBLICATIONS 2025. All Rights Reserved

PULSE OXIMETRY (SPO2 MEASUREMENT)

- Oxygenated hemoglobin (HbO2) absorbs more infrared light, while deoxygenated hemoglobin (Hb) absorbs more red light.
- By measuring the ratio of red and infrared light absorption, the sensor can determine the SpO2 level.

KEY FEATURES

- **Integrated:** The MAX30102 combines LEDs, a photo detector, optical elements, and low- noise electronics on a single chip.
- **Low Power:** It operates on a single 1.8V power supply and a separate 3.3V power supply for the LEDs.
- **I2C Communication:** It uses a standard I2C-compatible interface for communication with a microcontroller.
- Software Shutdown: The module can be shut down through software, reducing standby current.

APPLICATIONS

- Wearable devices (smart watches, fitness trackers).
- Mobile phones.
- Medical devices.

```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <PubSubClient.h>
#include <Wire.h>
#include "MAX30105.h"

// WiFi Credentials
#define WIFI_SSID "OPPO A16e"
#define WIFI_PASSWORD "i20pzano"

// HiveMQ Cloud MQTT Broker
#define MQTT_BROKER "796567340bfe4c9fb081137890c1b46f.s1.eu.hivemq.cloud"
```


Page 1-15 © MANTECH PUBLICATIONS 2025. All Rights Reserved

```
#define MQTT_PORT 8883  // TLS Port
#define MQTT_USERNAME "rajjamdar"  // Replace with your HiveMQ credentials
#define MQTT_PASSWORD "1212Raj+"  // Replace with your HiveMQ credentials
#define MQTT_TOPIC "esp32/health"

// Secure WiFi Client for TLS WiFiClientSecure espClient; PubSubClient client(espClient);
MAX30105 particleSensor;

// ECG Sensor Pin
const int ECG_PIN = 34;

// Buffer Size
const int bufferSize = 100; unsigned long lastPeakTime = 0; int beatsPerMinute = 0;

// ECG Moving Average Filter const int windowSize = 10;
int ecgBuffer[windowSize]; int bufferIndex = 0;
unsigned long lastLoopTime = 0; // Time tracker for non-blocking delay void
connectToWiFi() {
Serial.print("Connecting to WiFi..."); WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
while (WiFi.status() != WL_CONNECTED) { delay(500);
Serial.print(".");
}
Serial.println("\nWiFi Connected!");
}

void connectToMQTT() { while (!client.connected()) {

Serial.print("Connecting to MQTT...");
if (client.connect("ESP32Client", MQTT_USERNAME, MQTT_PASSWORD)) {
Serial.println("Connected to HiveMQ!");
} else { Serial.print("Failed, rc="); Serial.print(client.state());
Serial.println(" Retrying..."); delay(5000);
}
```


Page 1-15 © MANTECH PUBLICATIONS 2025. All Rights Reserved

```
}
}

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

// Secure Connection
espClient.setInsecure();  // Use this if you don’t want to verify SSL certificates

client.setServer(MQTT_BROKER, MQTT_PORT); connectToMQTT();

if (!particleSensor.begin(Wire, I2C_SPEED_STANDARD)) { Serial.println("MAX30102 not found!");
while (1);
}

particleSensor.setup();                    particleSensor.setPulseAmplitudeRed(0x0A);
particleSensor.setPulseAmplitudeIR(0x0F);

pinMode(ECG_PIN, INPUT);
for (int i = 0; i < windowSize; i++) { ecgBuffer[i] = 0;
}
}

void loop() {
unsigned long currentTime = millis();

// Non-blocking delay (500ms)
if (currentTime - lastLoopTime >= 500) { lastLoopTime = currentTime;

if (!client.connected()) { connectToMQTT();
}
client.loop();
```


Page 1-15 © MANTECH PUBLICATIONS 2025. All Rights Reserved

```
int redBuffer[bufferSize], irBuffer[bufferSize];

for (int i = 0; i < bufferSize; i++) { redBuffer[i] = particleSensor.getRed(); irBuffer[i] = particleSensor.getIR();
}

float redAC = calculateAC(redBuffer, bufferSize); float redDC = calculateDC(redBuffer, bufferSize); float irAC = calculateAC(irBuffer, bufferSize); float irDC = calculateDC(irBuffer, bufferSize);

float R = (redAC / redDC) / (irAC / irDC); float spo2 = 110 - 25 * R;
beatsPerMinute = calculateHeartRate(irBuffer, bufferSize);

int ecgRawValue = analogRead(ECG_PIN);
int smoothedECG = applyMovingAverageFilter(ecgRawValue);

// ECG waveform analysis
int prInterval = calculatePRInterval(); int qtInterval = calculateQTInterval();
int qrsDuration = calculateQRSDuration(); float stSegment = analyzeSTSegment();

String payload = "{";
payload += "\"spo2\":" + String(spo2) + ",";
payload += "\"heart_rate\":" + String(beatsPerMinute) + ","; payload += "\"ecg\":" + String(smoothedECG) + ","; payload += "\"pr_interval\":" + String(prInterval) + ","; payload += "\"qt_interval\":" + String(qtInterval) + ","; payload += "\"qrs_duration\":" + String(qrsDuration) + ","; payload += "\"st_segment\":" + String(stSegment);
payload += "}";

client.publish(MQTT_TOPIC, payload.c_str()); Serial.println("Data sent: " + payload);
}
}
float calculateAC(int *buffer, int size) {
int maxVal = buffer[0], minVal = buffer[0]; for (int i = 1; i < size; i++) {
```


Page 1-15 © MANTECH PUBLICATIONS 2025. All Rights Reserved

```
if (buffer[i] > maxVal) maxVal = buffer[i]; if (buffer[i] < minVal) minVal = buffer[i];
}
return maxVal - minVal;
}

float calculateDC(int *buffer, int size) { long sum = 0;
for (int i = 0; i < size; i++) { sum += buffer[i];

}
return (float)sum / size;
}

int calculateHeartRate(int *buffer, int size) { int peakCount = 0;
unsigned long currentTime = millis(), timeBetweenPeaks = 0;

for (int i = 1; i < size - 1; i++) {
if (buffer[i] > buffer[i - 1] && buffer[i] > buffer[i + 1]) { peakCount++;
if (peakCount > 1) {
timeBetweenPeaks = (currentTime - lastPeakTime) / (peakCount - 1);
}
}
}

lastPeakTime = currentTime;
return (timeBetweenPeaks > 0) ? (60000 / timeBetweenPeaks) : 0;
}

int applyMovingAverageFilter(int newSample) { ecgBuffer[bufferIndex] = newSample;
bufferIndex = (bufferIndex + 1) % windowSize;

long sum = 0;
for (int i = 0; i < windowSize; i++) { sum += ecgBuffer[i];
}
```


Page 1-15 © MANTECH PUBLICATIONS 2025. All Rights Reserved

```
return sum / windowSize;
}

int calculatePRInterval() {
// Logic to calculate PR Interval (start of P-wave to start of QRS) return 120;  // Placeholder value
}

int calculateQTInterval() {
// Logic to calculate QT Interval (start of QRS to end of T-wave) return 350;  // Placeholder value
}

int calculateQRSDuration() {
// Logic to calculate QRS Complex Duration return 90;  // Placeholder value
}

float analyzeSTSegment() {
// Logic to analyze ST Segment (between S-wave and T-wave) return 0.15;  // Placeholder value
}
```

CONCLUSION

A Smart Healthcare System integrates advanced technologies such as AI, IoT, and big data to enhance patient care, improve efficiency, and reduce medical errors. By leveraging real-time monitoring, predictive analytics, and automation, these systems provide personalized treatment, early disease detection, and seamless communication between healthcare providers and patients. In conclusion, smart healthcare systems revolutionize traditional healthcare by making it more accessible, efficient, and data-driven. While challenges like data security, system integration, and cost remain, continuous advancements in technology will further enhance patient outcomes and streamline healthcare services. The future of healthcare lies in intelligent, connected, and patient- centric solutions.

Page 1-15 © MANTECH PUBLICATIONS 2025. All Rights Reserved

REFERENCES

1. A course in Electrical and Electronic Measurements and Instrumentation-By A.K. Sawhney
2. A course in Electrical and Electronic Measurements and Instrumentation-By J.B. Gupta
3. Electronic Measurements and Instrumentation- By. R.K.Rajput
4. https://www.heart.org (American Heart Association (AHA))
5. https://www.escardio.org/ (European Society of Cardiology (ESC))
6. https://www.nih.gov/ (National Institutes of Health (NIH)) 4.https://www.cdc.gov/ (Centers for Disease Control and Prevention (CDC))
7. https://www.who.int/ (World Health Organization (WHO))

Page 1-15 © MANTECH PUBLICATIONS 2025. All Rights Reserved