Remote Heart Rate and ECG Monitoring System

PROJECT REPORT ON

Remote Heart Rate and ECG Monitoring System

UNDER THE GUIDANCE OF

Ms. KRUPA KAMDAR

SUBMITTED BY

Shubham Singh

SEAT NO. TYCS45

UNIVERSITY OF MUMBAI

T.Y.B.Sc (COMPUTER SCIENCE)

ACADEMIC YEAR: 2022-2023

BHARTIYA VIDYA BHAVAN'S

MM COLLEGE OF ARTS, NM COLLEGE OF COMMERCE &

HRJ COLLEGE OF COMMERCE

BHAVAN'S COLLEGE

MUNSHI NAGAR, ANDHERI (WEST),

MUMBAI – 400058

BHARTIYA VIDYA BHAVAN’S

MM COLLEGE OF ARTS, NM COLLEGE OF COMMERCE & HRJ COLLEGE OF COMMERCE

BHAVAN’S COLLEGE MUNSHI NAGAR,
ANDHERI (WEST),

MUMBAI – 400058

BHAVAN’S COLLEGE, ANDHERI (W)

CERTIFICATE

This is to certify that **Mr. Shubham Singh, Seat No. TYCS45** of the class **T. Y. B. Sc. Computer Science** has satisfactorily completed the practical course in **Group 6** as prescribed by the University of Mumbai during the academic year **2022 – 23.**

Signature
Staff in Charge

Signature
Computer Science Coordinators

Signature
Examiner

College Stamp

INDEX

Remote Heart Rate and ECG Monitoring System

1. Abstract and keywordsword

Abstract :

- Health monitoring and its related technologies is an attractive research area. The electrocardiogram (ECG) has always been a popular measurement scheme to assess and diagnose cardiovascular diseases (CVDs).
- The number of ECG monitoring systems in the literature is expanding exponentially. Hence, it is very hard for researchers and healthcare experts to choose, compare, and evaluate systems that serve their needs and fulfill the monitoring requirements.
- Understanding ECG monitoring systems' components, contexts, features, and challenges. Hence, a generic architectural model for ECG monitoring systems is proposed, an extensive analysis of ECG monitoring systems' value chain is conducted, and is presented, highlighting challenges and current trends.
- Finally, we identify key challenges and emphasize the importance of smart monitoring systems that leverage new technologies, including deep learning, artificial intelligence (AI), Big Data and Internet of Things (IoT), to provide efficient, cost-aware, and fully connected monitoring systems.

Keywords :

1. IOT
2. ESP32
3. AD8232
4. Arduino IDE
5. Web Programming
6. Jumper Wires
7. ECG monitoring system
8. smart monitoring,
9. heart diseases,
10. cardiovascular diseases,

2. Introduction

- ECG (Electrocardiogram) is a non-invasive medical test that measures the electrical activity of the heart. It is a valuable diagnostic tool that helps healthcare professionals assess the heart's health and identify any abnormalities in its function.
- AD8232 is a single-lead, heart rate monitor front-end IC (integrated circuit) that allows the development of wearable, battery-powered, and low-cost ECG solutions. It offers high signal quality and excellent ECG waveform extraction for a wide range of applications.
- The AD8232 integrates instrumentation amplifiers, right-leg drive, and leads-off detection to ensure accurate and reliable ECG measurement. It operates from a single power supply and consumes very low power, making it an ideal solution for portable and wearable applications.

2.1 Problem Statement

- The last decade has witnessed an increasing number of deaths caused by chronic and cardiovascular diseases (CVDs) in all countries across the world.
- CVDs are disorders affecting the blood vessels and the heart. CVDs involving the blood vessels are known as vascular diseases, such as coronary artery disease.
- Those involving the heart include heart failure, cardiomyopathy, rheumatic heart diseases, stroke, heart attack, and arrhythmias.

2.2 Literature Review/Description of Present System

- When the ECG machine displays an unusual pattern, there can be multiple unrelated reasons, including a normal variant. A doctor needs to do more detailed investigation, including other tests (e.g. echocardiogram), to sort things out.
- False negative is probably the biggest concern with ECG. For some heart patients, the EKG may be entirely normal and yet their conditions should be reflected in the ECG. The reasoning behind this is not well understood. A good ECG reading does not preclude having the underlying heart disease and other symptoms, such as chest pain, must be taken into account and further evaluation may be required.
- Not all heart problems will show up on an ECG. A prime example is vulnerable plaque (a form of atheroma). Vulnerable plaque is a fast growing deposit or degenerative accumulation of lipid-containing plaques on the innermost layer of the wall of an artery. Because artery walls typically enlarge in response to enlarging plaques, they do not affect blood flow and cannot be detected even in a cardiac stress ECG test. Yet vulnerable plaque is a major cause of heart attacks.

2.3 Background

An ECG has its limitations:

- The ECG reveals the heart rate and rhythm only during the few seconds it takes to record the tracing.
- If an arrhythmia (heart rhythm irregularity) occurs only intermittently, an ECG might not pick it up, and ambulatory monitoring may be required.
- The ECG is often normal or nearly normal with many types of heart disease, such as coronary artery disease.
- Sometimes, abnormalities that appear on the ECG turn out to have no medical significance after a thorough evaluation is done.

2.4 Aim & Objectives

- Continuous heart rate monitoring and immediate heartbeat detection are primary concerns in contemporary healthcare.
- Experimental evidence has shown that many of the CVDs could be better diagnosed, controlled, and prevented through continuous monitoring, as well as analysis of electrocardiogram (ECG) signals.
- Hence, the monitoring of physiological signals, such as electrocardiogram (ECG) signals, offers a new holistic paradigm for the assessment of CVDs, supporting disease control and prevention.
- With advances in sensor technology, communication infrastructure, data processing, and modeling as well as analytics algorithms the risk of impairments could be better addressed more than ever done before.
- This, in turn, would introduce a new era of smart, proactive healthcare especially with the great challenge of limited medical resources.

2.5 Project Motivation

Early diagnosis and detection of heart disease is important to avoid sudden death due to heart attack or cardiac arrest. Electrocardiogram (ECG) sensor is a device commonly used by cardiologists to check for abnormal heart rhythm and signs of potential heart disease quickly and without intervention.

3. Description of Proposed Work

- The Embedded technology has entered almost in all aspects of day-to-day life, and the healthcare field is no exception for that the requirement for fully-equipped hospitals and diagnostic centers growing day by day as people are becoming more unaware of their health problems.
- An ECG signal can trace various physiological and abnormal conditions of the heart. This heart monitoring system also helps to inform the person whether he/she has any heart diseases or not.
- This is done by checking the heart beat level. In this system Atmega controller is used to scan ECG signal and search for pattern in common range, if the pattern will be in common range then it gives the report of being normal if it is found that the is not in common range then the person is suffering from some kind of heart disease.

3.1 Number of Modules

1. AD8232 ECG Sensor Module:

   This module is used to measure the electrical activity of theheart and generate an ECG signal.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. This design allows for an ultralow power analog-to-digital converter (ADC) or an embedded microcontroller to acquire the output signal easily.

2. ESP32 Development Board:

   This board is used to process the ECG signal from the AD8232 module and send the data to the internet through Wi-Fi. It is a low cost and low power development board by espressif systems. The board can easily connect to a wifi or create it’s own wifi.

3. Wi-Fi Module:

   This module is used to connect the ESP32 to the internet and send the data to a web server. the ESP32 can easily connect to a Wi-Fi network to connect to the internet (station mode), or create its own Wi-Fi wireless network (access point mode) so other devices can connect to it this is essential for IoT and Home Automation projects—you can have multiple devices communicating with each other using their Wi-Fi capabilities;

4. HTML/CSS/JavaScript:

   These web technologies are used to create a web page that displays the ECG signal in real-time. The web page continously requests new data from the esp32 server for ecg signals.

5. Web Server:

   A web server is used to host the web page and receive the data from the ESP32. The web server is mobile responsive and can be accessed with any device that as a browser on the local network

6. MDNS :

mDNS is a multicast UDP service that is used to provide local network service and host discovery. communication between the ESP32 and the web browser.
In computer networking, the multicast DNS (mDNS) protocol resolves hostnames to IP addresses within small networks that do not include a local name server. It is a zero-configuration service, using essentially the same programming interfaces, packet formats and operating semantics as unicast Domain Name System (DNS).

8. Chart.js:

Chart.js is a JavaScript library that can be used to create interactive charts and graphs on a web page. It is one of the simplest visualization libraries for JavaScript, and comes with the following built-in chart types:

9. File Download Module:

The module helps to download the raw Ecg signals file generated from the esp32 and the ecg sensor. The file could be later used to inference the health or use ML to predict the health of the patient.

3.2 Algorithm

The Algorithm Used is for the backend and the esp32 :

1. Connect the Esp32 to Wifi
2. Check for the presence of the Sensor module.
3. Create a temporary storage buffer of ten seconds for raw ecg signals
4. Create a timer of 500 Hertz :
   for each timer interupt:
   check for the current time in milli seconds
   create the index to store the sensor data
   read the sensor data from the ECG sensor
5. Create a esp32 web server and do request mappings for url parameters
6. create a mDNS server for browsers to connect to the esp32 server with local domain name.
7. On request for raw data :
   check if prevoius data time is present :
   if present send a data up to date response.
   Else :
   send the resent data present to the client browser.

For Browser.

1. Load the chart js and request library’s
2. check for the presence of a ecg server on the lan network
3. if server is available fetch the raw signals from the server with get requests.
   Store the data in an array for representation of graph and download of data with file.
4. Using the chart js library draw the graph on the browser window.
5. On user download button clicked:
   Convert the stored raw ecg data array to blob file data
   create an anchor tag with the href ointing to the creted data file
   click on the link with program and store the data in a csv file

3.3 Working

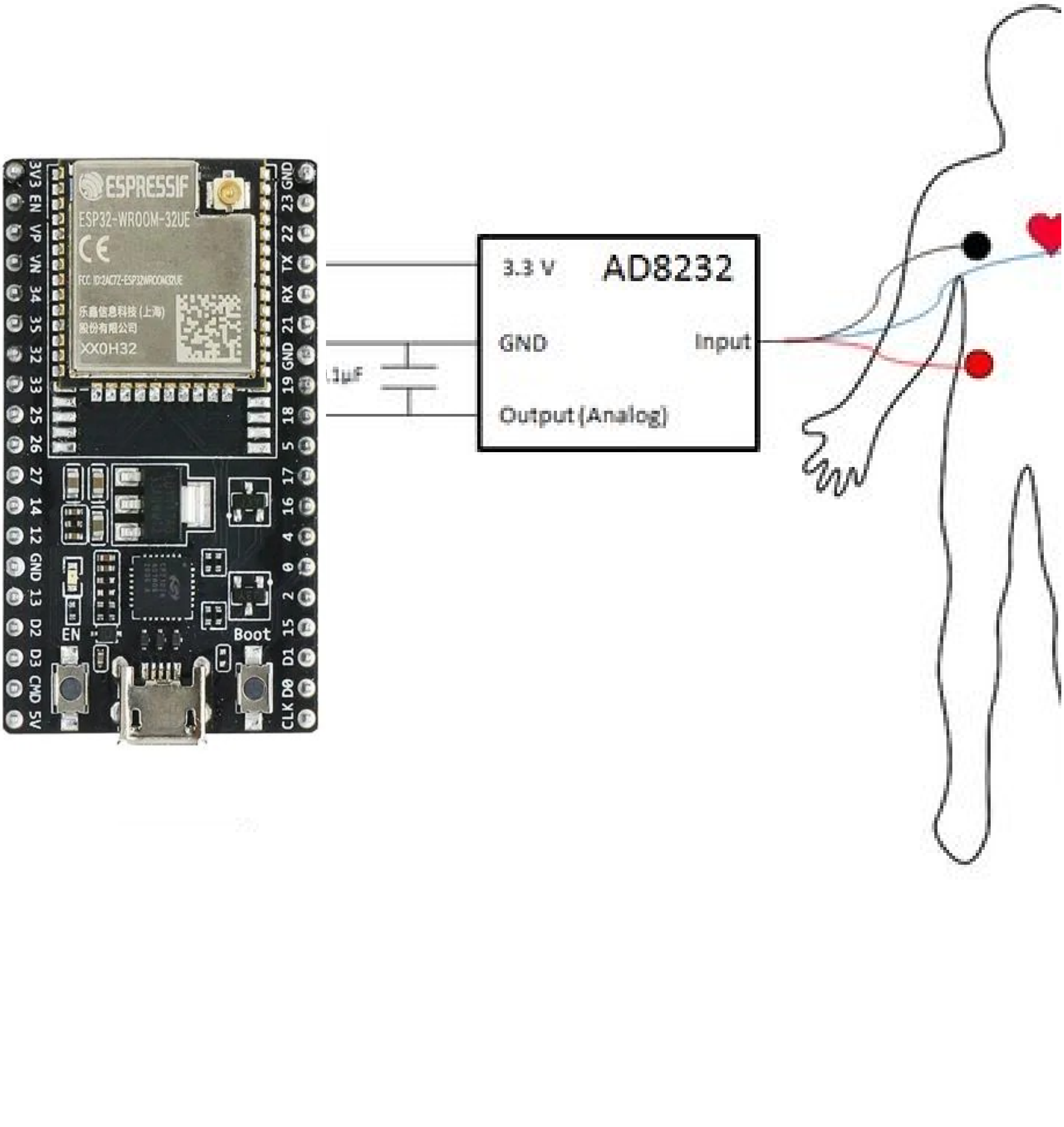
Following components are used for this project.

1. Esp 32 Development Board
2. Jumper Wires
3. Ad8232 ECG Sensor
4. Buzzer
5. ECG Electrodes
6. Resistors

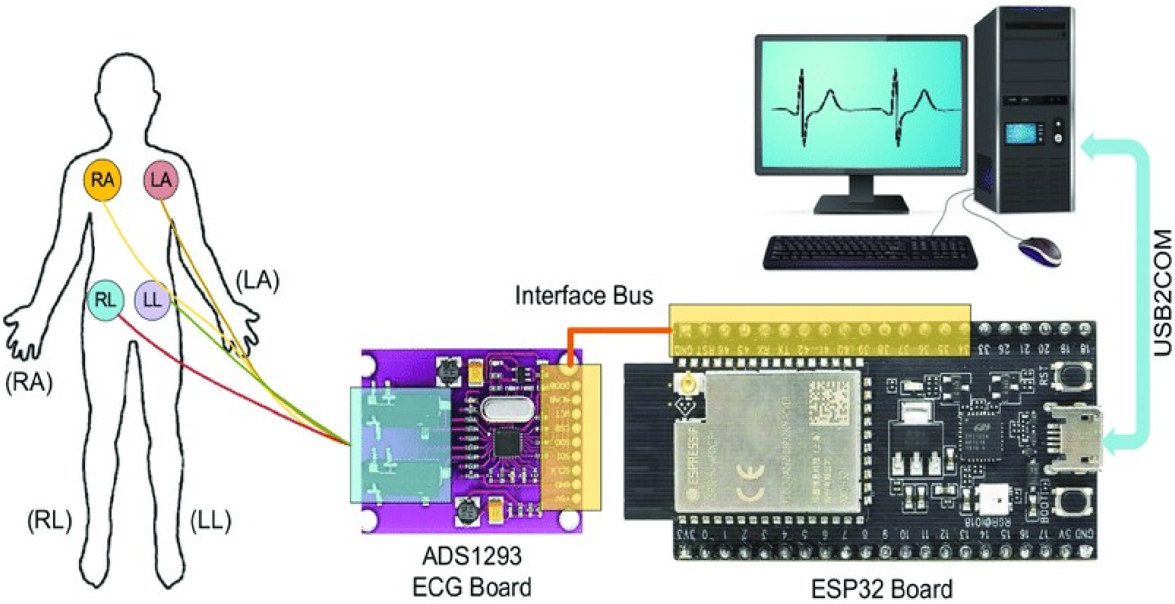
- The Iot projecct is used for measuring the ECG signals of a patient.
- This is achived by attaching ECG eletrodes to the patient at appropriate locations the attached electrodes are then connected to the sensor modlue named as ad8232.
- The sensor module is based on high sensitivity operationsal apmlifiers or OpAmps and noice cancellation with active network of resistors and capacitors.
- The AD8232 ECG sensor consists of three electrodes that are attached to the chest to measure the electrical signals produced by the heart. The output from the electrodes is then amplified by an instrumentation amplifier and filtered to remove any noise or interference.
- The filtered signal is then fed into the analog-to-digital converter (ADC) of the ESP32 development board, which converts the analog signal into a digital signal that can be processed by the board.
- The ESP32 development board is programmed to read the digital ECG signal and send it to a web server through Wi-Fi. The data can be transmitted using protocols such as HTTP, depending on the specific implementation.
- On the web server, the data can be analyzed and displayed using various web technologies such as HTML, CSS, JavaScript, and Chart.js. Real-time updates can be achieved by using a combination of server-side programming and client-side programming (using JavaScript and AJAX).
- The AD8232 ECG sensor measures the electrical activity of the heart, the ESP32 development board processes the ECG signal and transmits it to a web server, and the web server analyzes and displays the data in real-time using various web technologies. Such as Chart.js graph
- The data is then made available to the patient in a raw csv file for further utilization, analysis and experimentations.

3.4 <u>Design/Block diagram/flow chart/graph/deployment diagram</u>

- **Circuit Diagram**



- **Work Flow**



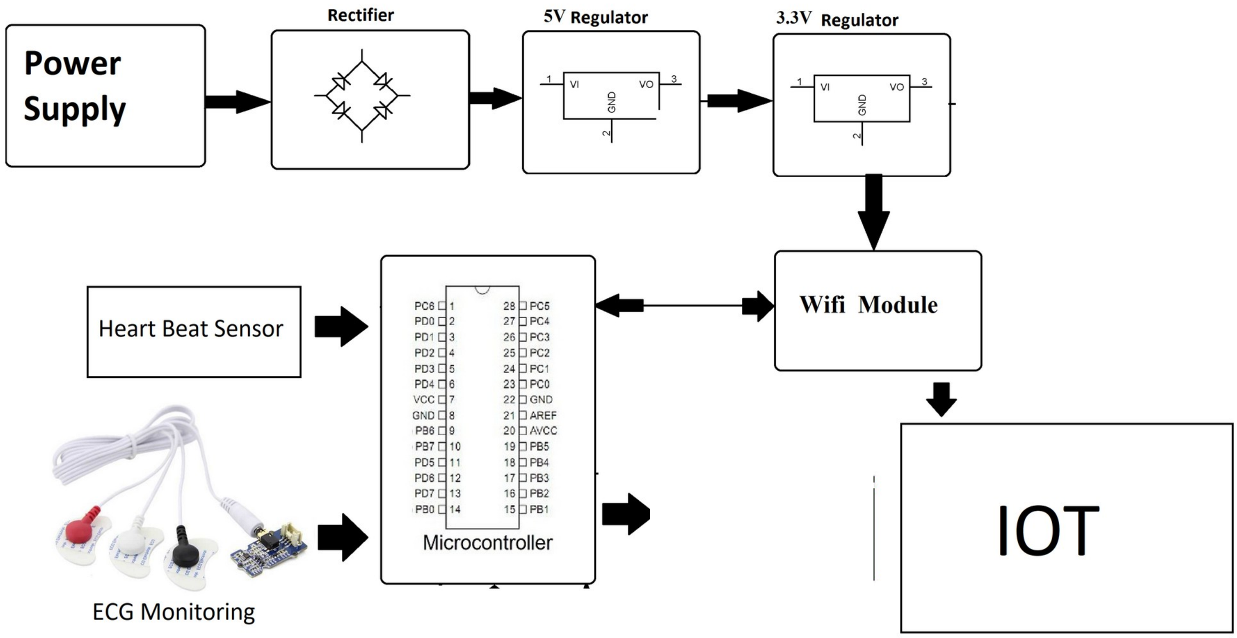
RA
LA
RL
LL
(LA)
(RA)
(RL)
(LL)
Interface Bus
ADS1293
ECG Board
ESP32 Board
USB2COM

Architectural Design

To build an ECG (electrocardiogram) monitoring system using the AD8232 sensor and ESP32, you will need the following architecture components:

1. Power Supply: The AD8232 sensor module and the ESP32 development board both require a stable power supply. A USB cable connected to a computer or a USB wall adapter can be used to power the system.

2. ECG Electrodes: These are small adhesive pads that are placed on the chest to pick up the electrical signals produced by the heart.



3. AD8232 Sensor Module: This is a small module that contains the AD8232 integrated circuit, which is designed to read ECG signals from the body. The module also includes an instrumentation amplifier, a high-pass filter, and a low-pass filter.

4. ESP32 Development Board: This is a powerful microcontroller board that has built-in Wi-Fi and Bluetooth capabilities. It can be programmed using the Arduino IDE and is compatible with a wide range of sensors and other peripherals.

5. Amplifier Circuit: In order to amplify the signals from the ECG electrodes, an additional amplifier circuit may be required. This can be a simple operational amplifier circuit that is connected between the electrodes and the AD8232 sensor module.

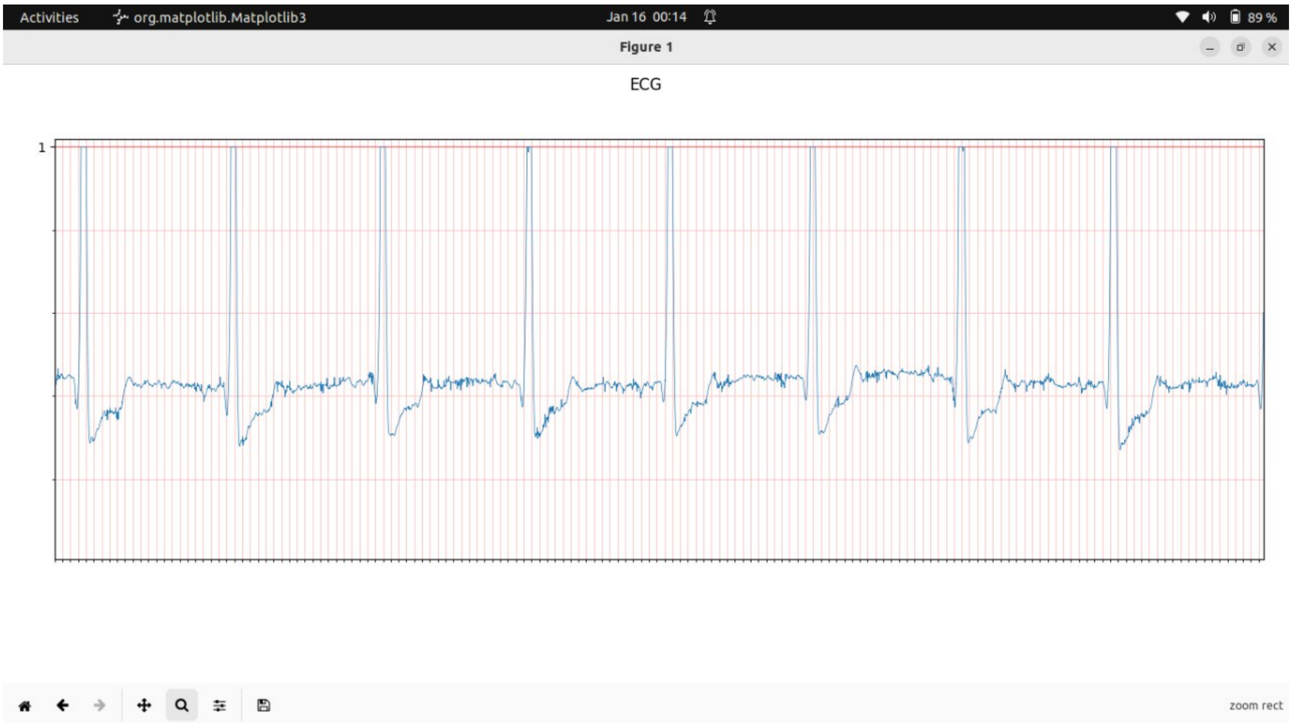
6. Bluetooth or Wi-Fi Module: If you want to transmit the ECG signals wirelessly, you will need a Bluetooth or Wi-Fi module that is compatible with the ESP32 development board.

7. Display: A display can be used to show the ECG waveform in real-time. This can be an OLED display or an LCD display that is connected to the ESP32 development board.

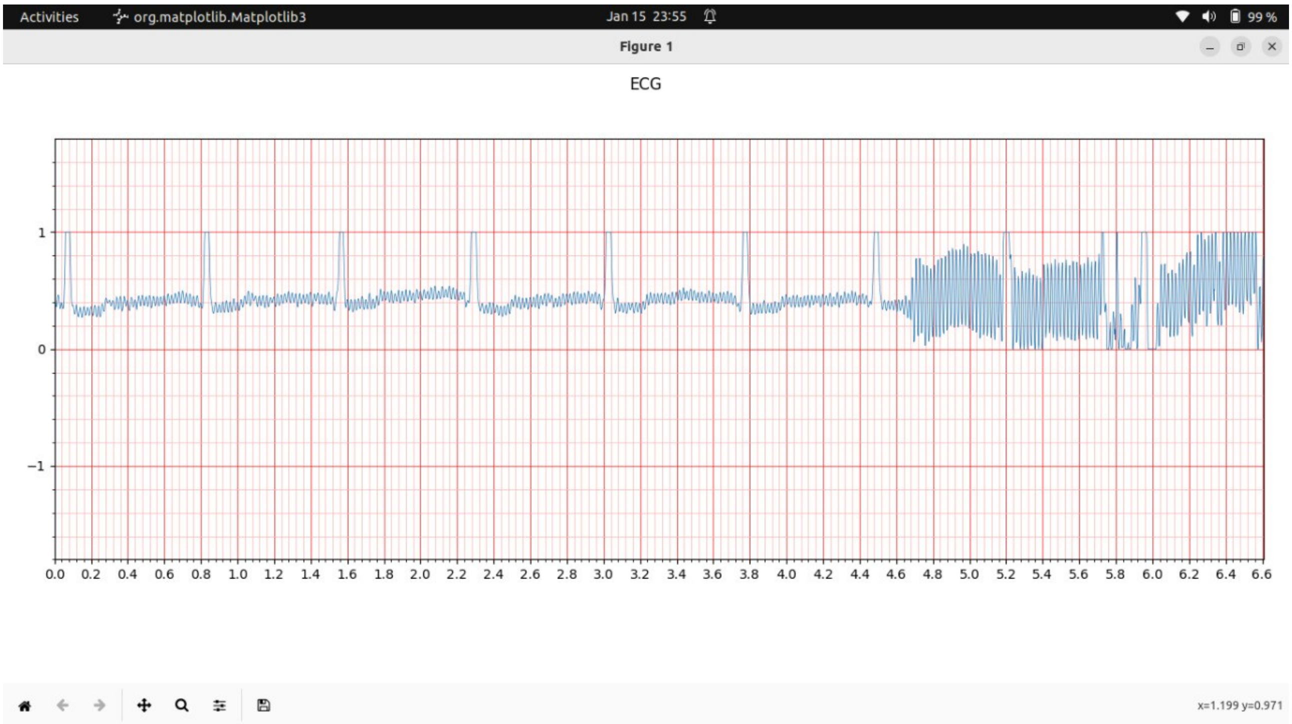
Overall, the architecture components for an ECG monitoring system using the AD8232 sensor and ESP32 consist of the sensor module, the development board, power supply, electrodes, amplifier circuit, wireless module, and display. With these components, you can build a reliable and accurate ECG monitoring system that can be used for medical research, fitness tracking, and more.

Experimental results and evaluation

The performance of an ECG monitoring system using the AD8232 sensor with ESP32 is been evaluated based on several factors such as accuracy, reliability, and usability.



Accuracy: The accuracy of the ECG monitoring system can be evaluated by comparing the measured ECG signals with a reference signal obtained from a professional medical ECG device. The system should be able to accurately detect the different waves and intervals of the ECG signal such as the P wave, QRS complex, and T wave. The accuracy can be quantified using metrics such as sensitivity, specificity, and positive predictive value.



Noise rejection: The AD8232 sensor has a built-in instrumentation amplifier and filters that help to reject noise and interference from the ECG signal. The system should be able to effectively filter out noise from sources such as electromagnetic interference (EMI) and motion artifacts. The noise rejection can be evaluated by introducing different types of noise and measuring the system's ability to reject them.

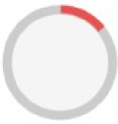
Usability: The usability of the ECG monitoring system can be evaluated based on factors such as ease of use, comfort, and portability. The system should be easy to use and comfortable for the user to wear for extended periods of time. The portability can be evaluated by measuring the weight and size of the system and comparing it to other similar devices.

Wireless transmission: The system includes a wireless module for transmitting the ECG signals to a remote device, the reliability of the wireless transmission should be evaluated. This can be done by measuring the signal strength, latency, and packet loss rate.

3.5 Plagiarism report

Dupli Checker

PLAGIARISM SCAN REPORT



14%
Plagiarised

86%
Unique

Date	2023-03-04
Words	992
Characters	8125

Content Checked For Plagiarism

Remote Heart Rate and ECG Monitoring System

1. Abstract and keywords

Keywords :

1. IOT
2. ESP32
3. AD8232
4. Arduino IDE
5. Web Programming
6. Jumper Wires
7. ECG monitoring system
8. smart monitoring,
9. heart diseases,
10.cardiovascular diseases,

3.2 Algorithm
The Algorithm Used is for the backend and the esp32 :
1. Connect the Esp32 to Wifi
2. Check for the presence of the Sensor module.
3. Create a temporary storage buffer of ten seconds for raw ecg signals
4. Create a timer of 500 Hertz :
for each timer interupt:
check for the current time in milli seconds
create the index to store the sensor data
read the sensor data from the ECG sensor

5. Create a esp32 web server and do request mappings for url parameters
6. create a mDNS server for browsers to connect to the esp32 server with local domain name.

Matched Source

Similarity 8%

Title:Weather on the Web With Sense Hat : 8 Steps - Instructables

Weather on the Web With Sense Hat : 8 Steps - InstructablesInstructableshttps://www.instructables.com › Circuits › Raspberry PiInstructableshttps://www.instructables.com › Circuits › Raspberry PiThe following components are used for this project: Raspberry Pi 3 · Sense Hat · Sense Hat case. HDMI Cable. Monitor. Keyboard.

https://www.instructables.com/Weather-on-the-Web-With-Sense-Hat/

Similarity 6%

Title:EKG Sensor | Flinn Scientific

EKG Sensor | Flinn Scientifichttps://www.flinnsci.com › ekg-sensorhttps://www.flinnsci.com › ekg-sensorThis terrific sensor allows your students to quickly and easily measure the electrical signals produced by the heart, and at an affordable price.$196.39

https://www.flinnsci.com/ekg-sensor/tc1531/

Similarity 6%

Title:CLARA conceptual design report - IOPscience

CLARA conceptual design report - IOPsciencehttps://iopscience.iop.org › article › pdfhttps://iopscience.iop.org › article › pdfby JA Clarke · 2014 · Cited by 89 — The output from the electrodes is then sampled by optical clock pulses. (see section 5.5) by gating their amplitude in an electro-optic ...

https://iopscience.iop.org/article/10.1088/1748-0221/9/05/T05001/pdf

---

Similarity 4%

Title:Advanced WiFi Usage | Seeed Studio WikiAdvanced Wi-Fi Usage | Seeed Studio Wiki

Advanced WiFi Usage | Seeed Studio Wikihttps://wiki.seeedstudio.com › Wio-Terminal-Advanced...https://wiki.seeedstudio.com › Wio-Terminal-Advanced...Jan 13, 2023 — This wiki introduces some of the advanced library usage of the Wi-Fi such as HTTPClient, DNSServer and WebServer Libraries. With these libraries ...Advanced Wi-Fi Usage | Seeed Studio Wikihttps://wiki.seeedstudio.com › zh-CN › Wio-Terminal-...https://wiki.seeedstudio.com › zh-CN › Wio-Terminal-...Jan 11, 2022 — This wiki introduces some of the advanced library usage of the Wi-Fi such as HTTPClient, DNSServer and WebServer Libraries.

https://wiki.seeedstudio.com/Wio-Terminal-Advanced-WiFi/

3.6 Coding

Esp 32 Source Code

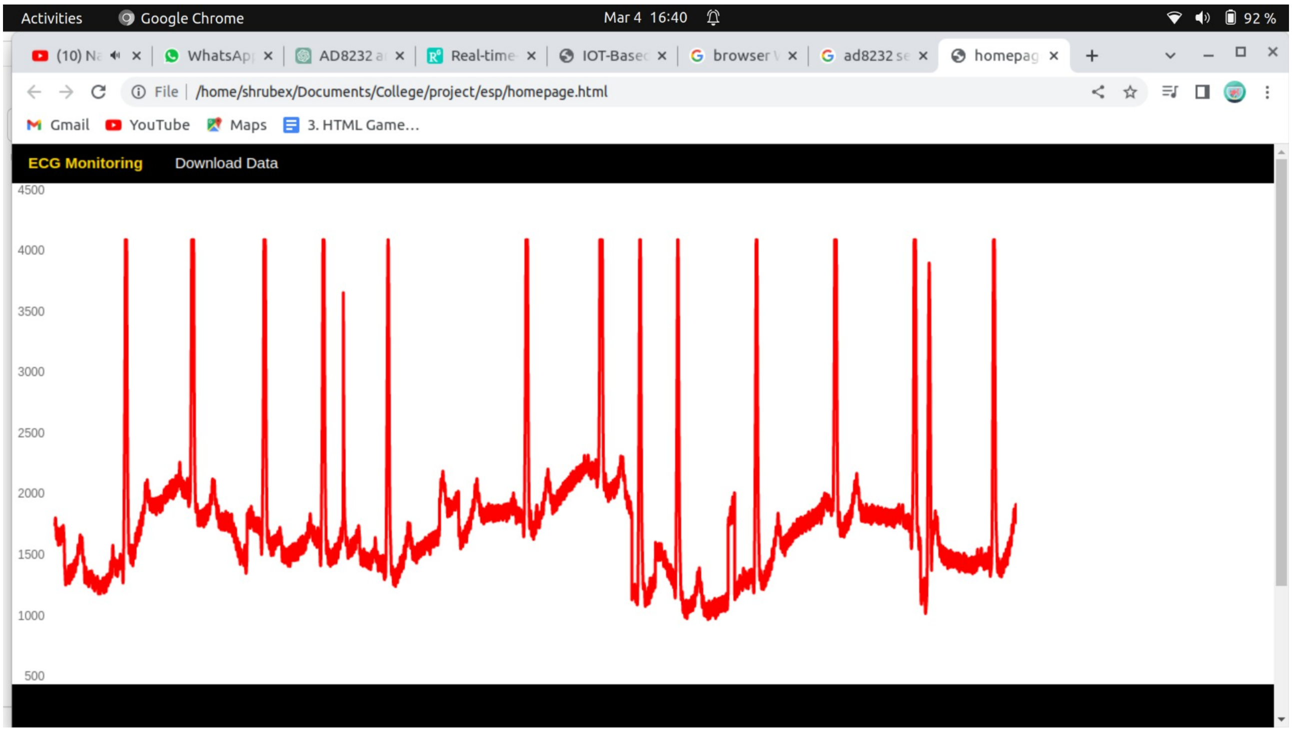
```
#include <WiFi.h>
#include <WiFiClient.h>
#include <WebServer.h>
#include <ESPmDNS.h>


#define LED 13
int IRAM_ATTR ECG_RAW[10][500];
// reading from heart sensor.
hw_timer_t *My_timer = NULL;
void IRAM_ATTR onTimer(){
  //Read value.
  int sec = millis() / 1000;
  int arrayInd = sec%10;
  int readingInd = (millis()%1000)/2;
  ECG_RAW[arrayInd][readingInd] = analogRead(34); // ecg pin attached to pin 34.
}

// web server
const char *ssid = "SS";
const char *password = "@12345678";

WebServer server(80);

const int led = 2;

void setup(void) {
  pinMode(led, OUTPUT);
  digitalWrite(led, 0);
  Serial.begin(115200);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  Serial.println("");

  //set up interupts for sensor reading.
  pinMode(LED, OUTPUT);
  My_timer = timerBegin(0, 80, true);
  timerAttachInterrupt(My_timer, &onTimer, true);
  timerAlarmWrite(My_timer, 2000, true);
  timerAlarmEnable(My_timer); //Just Enable

  // Wait for connection
  while (WiFi.status() != WL_CONNECTED) {
```

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

  Serial.println("");
  Serial.print("Connected to ");
  Serial.println(ssid);
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());

  if (MDNS.begin("ecg")) {
    Serial.println("MDNS responder started");
  }
  server.enableCORS(true);
  server.enableCrossOrigin(true);
  server.on("/", handleRoot);
  server.on("/raw", sendECG);
  server.on("/inline", []() {
    server.send(200, "text/plain", "this works as well");
  });
  server.onNotFound(handleNotFound);
  server.begin();
  Serial.println("HTTP server started");
}

void loop(void) {
  server.handleClient();
  delay(2);//allow the cpu to switch to other tasks
}

void handleRoot() {
  digitalWrite(led, 1);
  char temp[400];
  int sec = millis() / 1000;
  int min = sec / 60;
  int hr = min / 60;

  snprintf(temp, 400,

           "<html>\
  <head>\
    <meta http-equiv='refresh' content='1'/>\
    <title>ECG SYSTEM</title>\
    <style>\
      body { background-color: #cccccc; font-family: Arial, Helvetica, Sans-Serif; Color: #000088; }\
    </style>\
  </head>\
  <body>\
```

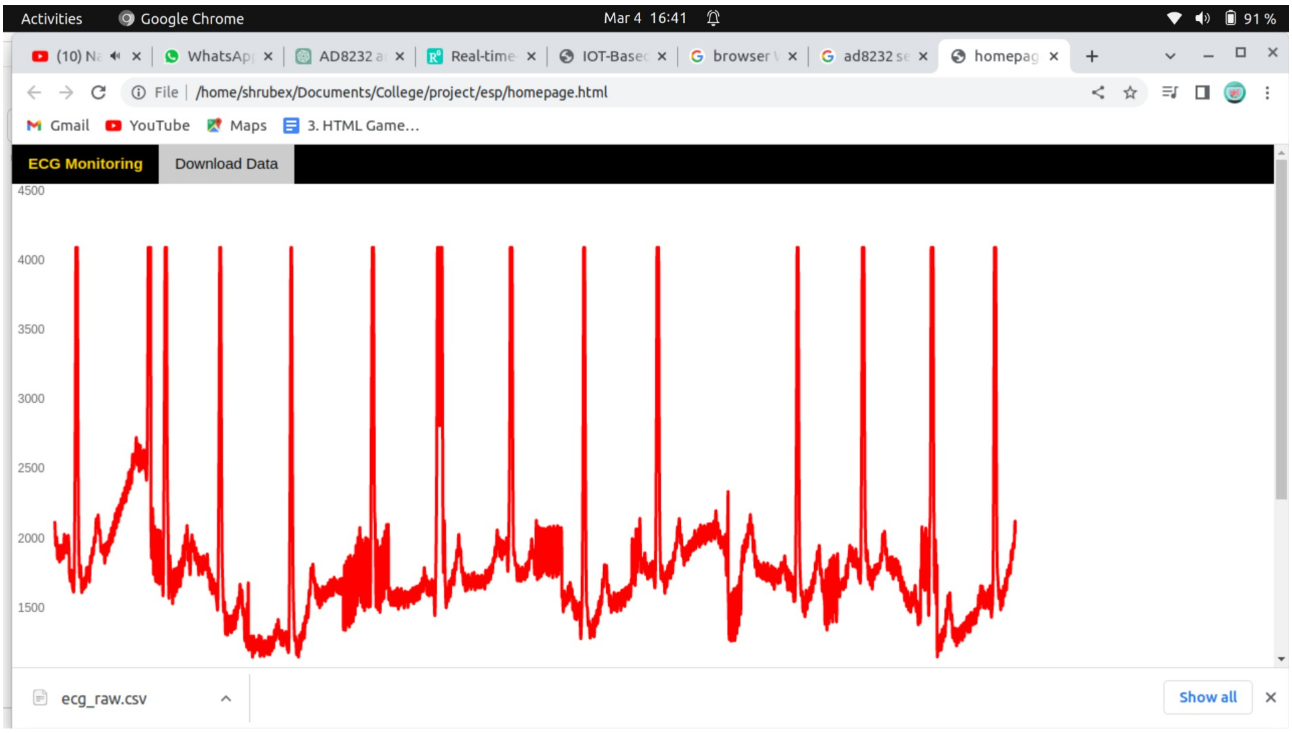
```
    <h1>ECG SENSOR</h1>\
    <p>Uptime: %02d:%02d:%02d</p>\
  </body>\
</html>",

           hr, min % 60, sec % 60
          );
  server.send(200, "text/html", temp);
  digitalWrite(led, 0);
}

void handleNotFound() {
  digitalWrite(led, 1);
  String message = "File Not Found\n\n";
  message += "URI: ";
  message += server.uri();
  message += "\nMethod: ";
  message += (server.method() == HTTP_GET) ? "GET" : "POST";
  message += "\nArguments: ";
  message += server.args();
  message += "\n";

  for (uint8_t i = 0; i < server.args(); i++) {
    message += " " + server.argName(i) + ": " + server.arg(i) + "\n";
  }

  server.send(404, "text/plain", message);
  digitalWrite(led, 0);
}


void sendECG(){
  int sec = millis() / 1000;
  int arrayInd = sec%10;
  int newData = 1;
  if(server.args() > 0){
    int prevSec =  server.arg(0).toInt();
    Serial.println(String(prevSec)+" <-> "+String(arrayInd));
    if(prevSec == arrayInd){
      Serial.println("same data "+String(sec));
      newData = 0;
    }else{
      arrayInd = (prevSec+1)%10;
    }
  }
  else if(arrayInd>0){ arrayInd--;}
  else{arrayInd = 9;}
  String out = "{\"sec\":"+String(arrayInd)+",";
```

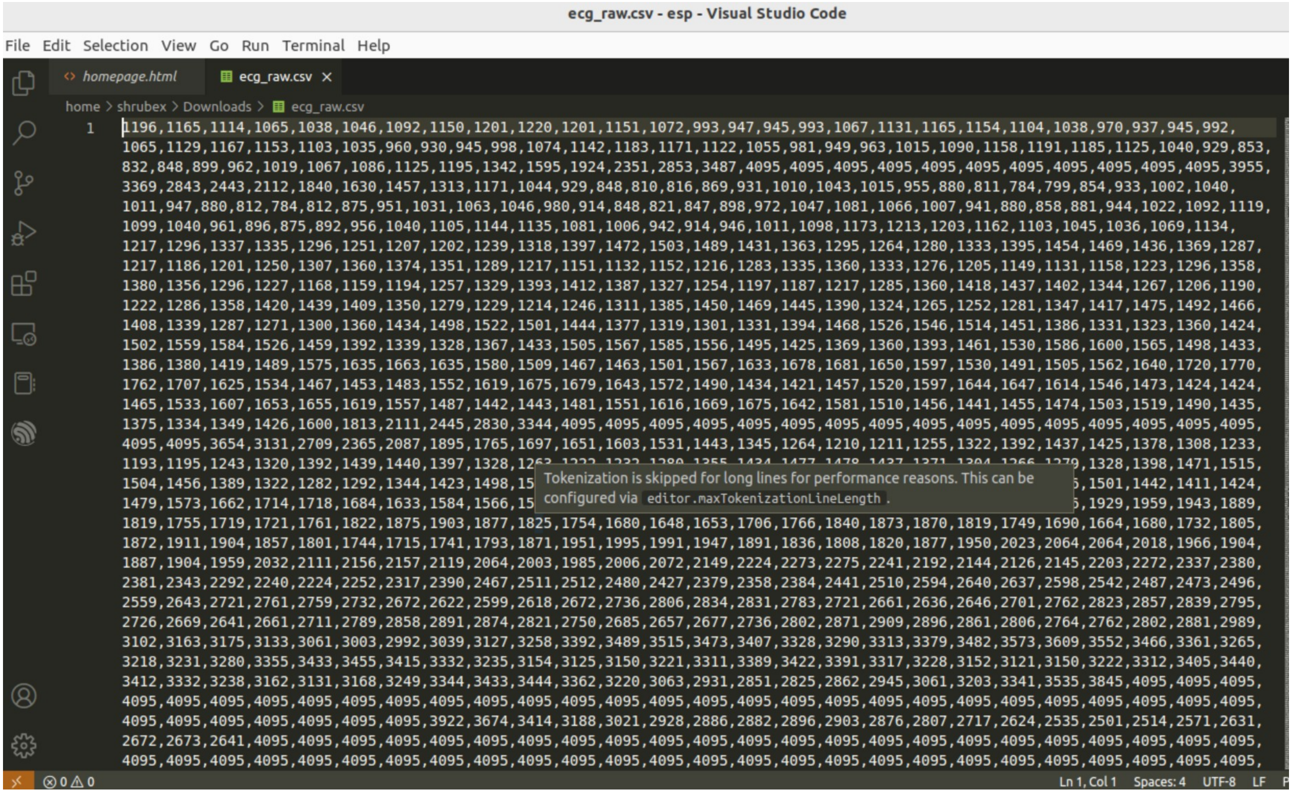
```
  out+="\"newData\":"+String(newData);
  if(newData==0){
    out+= "}";
  }else{
    out += ",\"data\":[";
    for(int i = 0 ; i < 499 ; i++){
      out += String(ECG_RAW[arrayInd][i])+",";
    }out += String(ECG_RAW[arrayInd][499])+"]}  ";
  }

  server.send(200, "application/json", out);
}
```

3.7 Screen Layouts



.




ecg_raw.csv - esp - Visual Studio Code
File Edit Selection View Go Run Terminal Help
homepage.html
ecg_raw.csv
home > shrubex > Downloads > ecg_raw.csv
Tokenization is skipped for long lines for performance reasons. This can be configured via editor.maxTokenizationLineLength.
Ln 1, Col 1 Spaces: 4 UTF-8 LF

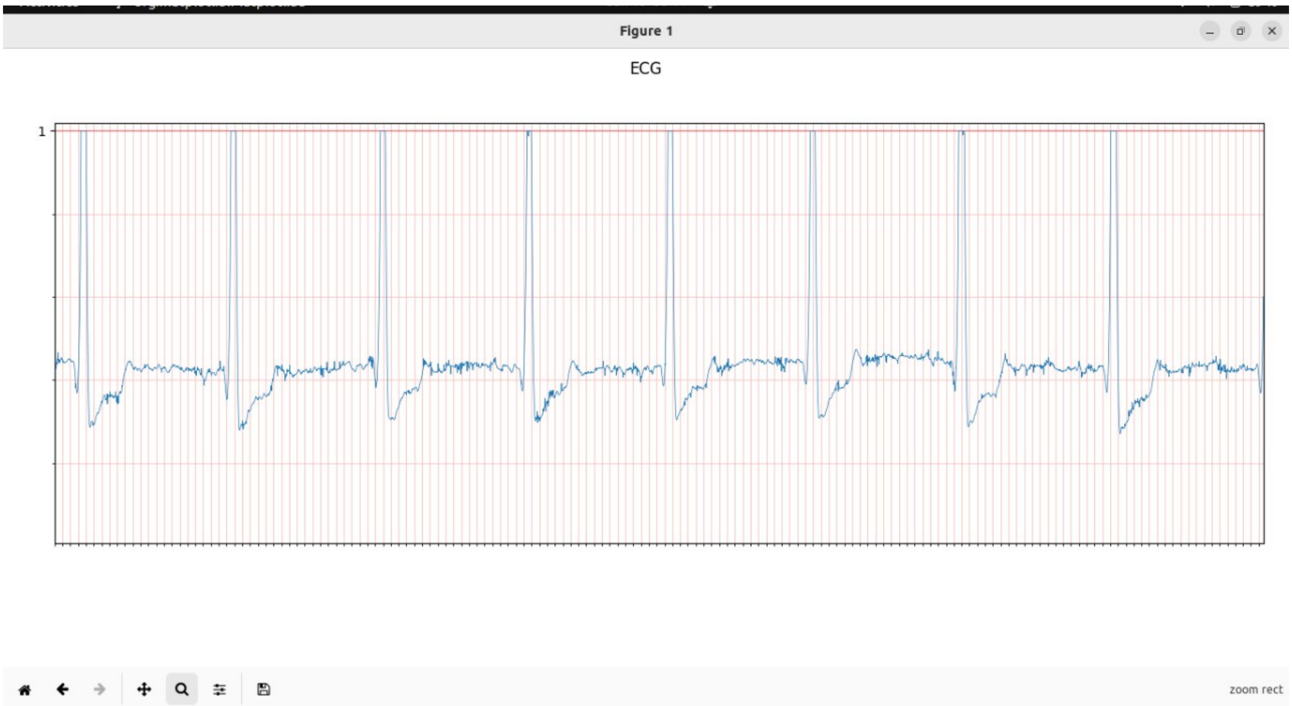
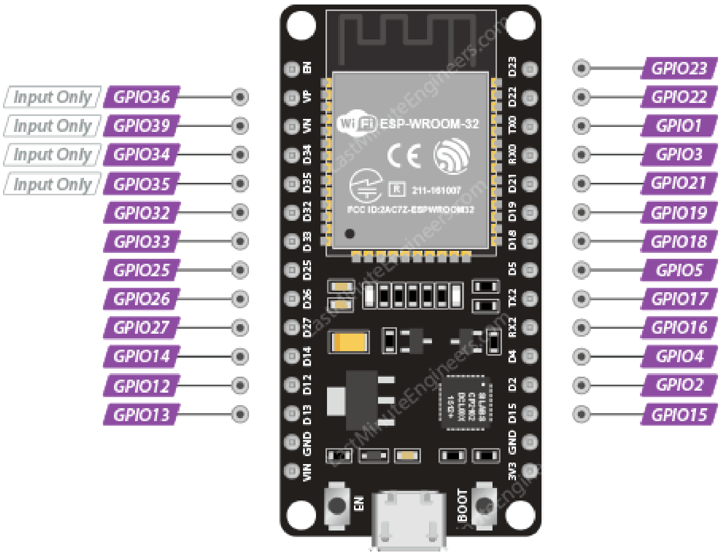


Figure 1
ECG
1
zoom rect

4. Technology/Language/Development Tools/Hardware/Software

Hardware

1. Esp32 Development Board

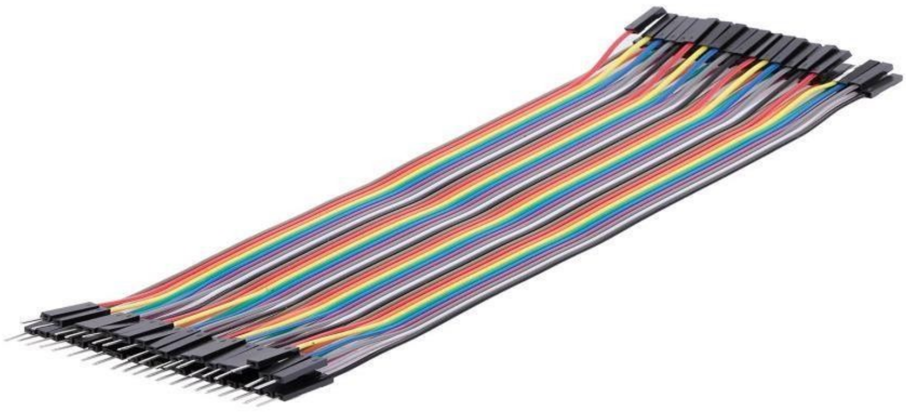


- The ESP32 development board is a type of microcontroller board based on the ESP32 system-on-a-chip (SoC), which is designed for use in IoT (Internet of Things) applications.
- The ESP32 SoC is a low-cost, low-power microcontroller with built-in Wi-Fi and Bluetooth capabilities, as well as a dual-core processor and a variety of other features that make it well-suited for IoT applications.

- ESP32 development boards are often used by developers, hobbyists, and engineers to prototype and develop IoT applications, as they provide an easy way to get started with the ESP32 platform.
- These development boards typically include a variety of useful features and components, such as USB ports for programming and power, voltage regulators, and GPIO pins for connecting sensors and other peripherals.
- They may also include additional features such as OLED displays, SD card slots, and other components that make it easier to build and test IoT applications.

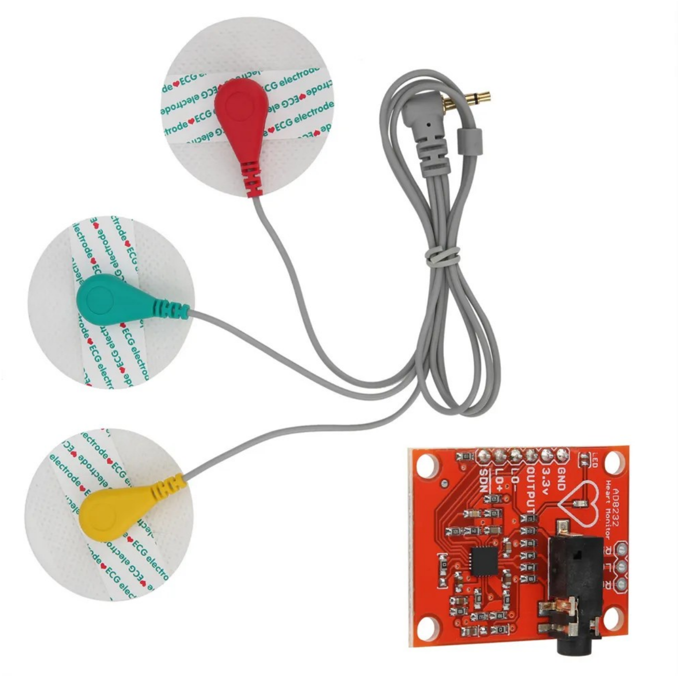
2. Jumper Wires

- Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points without soldering. Jumper wires are typically used with bread boards and other prototyping tools to make it easy to change a circuit as needed. Fairly simple. It doesn’t get much more basic than jumper wires.



3. AD8232 ECG sensor

- The AD8232 sensor is a specialized integrated circuit (IC) that is commonly used in the design of electrocardiogram (ECG) systems. This sensor is produced by Analog Devices, Inc., and is designed to amplify the tiny electrical signals generated by the human heart so that they can be accurately measured and analyzed.



- The AD8232 sensor is a two-stage instrumentation amplifier that is designed to provide high gain, low noise, and low input impedance for ECG applications. It includes several features that make it ideal for use in medical applications, including a low-pass filter to remove noise and interference, an adjustable gain amplifier, and a right-leg drive circuit to reduce common-mode interference.

- The AD8232 sensor is typically used in conjunction with a microcontroller or other data acquisition system, which receives the amplified ECG signal and processes it for display or further analysis. The sensor can be used in a wide range of applications, from simple heart rate monitors to more advanced diagnostic and monitoring systems used in hospitals and clinics.

Software Used

Arduino IDE

- The Arduino Integrated Development Environment (IDE) is a software application used to write, compile, and upload code to Arduino boards. Arduino is an open-source electronics platform based on easy-to-use hardware and software. It is designed for artists, designers, hobbyists, and anyone interested in creating interactive projects or prototypes.

- The Arduino IDE provides an easy-to-use interface for writing code in the Arduino programming language, which is based on C/C++. It includes a text editor for writing code, a compiler for converting code into machine language, and a bootloader for uploading the code to the Arduino board.

- The IDE also includes a serial monitor for debugging and testing code, and a library manager for installing and managing libraries of pre-written code. There are many libraries available for Arduino, which can be used to simplify programming tasks and add functionality to projects.

- The Arduino IDE is free and open source, and is available for Windows, Mac, and Linux operating systems. It is widely used in the maker and DIY communities, and is a popular tool for prototyping and creating a wide range of projects, from simple LED blinkers to complex robotic systems.

5. *Conclusion & Future scope*

Conclusion :

- Once the ECG data has been processed, it can be transmitted wirelessly to a remote server or mobile device using Wi-Fi or Bluetooth connectivity. The data can then be analyzed using specialized software or machine learning algorithms to detect patterns or trends in the data, and provide insights into heart health and activity.
- Overall, an AD8232 ECG sensor module with ESP32 project has the potential to revolutionize the way we monitor and diagnose heart health, and could have a significant impact on the field of medicine and healthcare.
- The latest trends in ECG monitoring systems will revolutionize the way ECG signals are collected and processed to give valuable insights serving various purposes while protecting patients' privacy and emotional health.
- Personalization and adaptation to various contexts as well as to various stakeholders will offer a new level of high-quality smart healthcare.
- Modern technologies will play a vital role in this radical transformation. These include radar cardiography, implants, robotics, steganography, and other AI technologies.

Future Scope

- Real-time monitoring: The project can be further developed to provide real-time monitoring of heart activity, which would allow healthcare professionals to quickly identify and respond to abnormal heart rhythms or other issues.
- Mobile health: With the proliferation of smartphones and wearable devices, the project could be adapted to provide mobile health monitoring and tracking, allowing individuals to monitor their heart activity and receive alerts or feedback on their health status.
- Machine learning: The project could be integrated with machine learning algorithms to provide more advanced analysis and insights into heart health and activity. Machine learning could be used to detect patterns or trends in the data that are not immediately apparent to human analysts, allowing for more accurate and personalized diagnoses and treatment plans.
- Remote diagnostics: The project could be used to provide remote diagnostics and consultations, allowing healthcare professionals to monitor and analyze ECG data from patients in remote or underserved areas.

6. References

"ECG Monitoring System using AD8232 and ESP32" by Anjali R. Patil and Prajakta S. Bhandarkar, International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 2020.

https://www.ijsrcseit.com/paper/CSEIT205165.pdf

"Real-Time Heart Monitoring System Using AD8232 and ESP32" by J. Daniel and J. Ponniah, International Journal of Innovative Research in Computer and Communication Engineering, 2019.

https://www.ijircce.com/upload/2019/april/63C_real.pdf

"Wireless ECG monitoring system using ESP32 and cloud-based server" by M. U. Khan and S. A. Alvi, 2019 IEEE 6th International Conference on Engineering Technologies and Applied Sciences (ICETAS).

https://ieeexplore.ieee.org/abstract/document/8953641

"Real-time ECG monitoring system with AD8232 sensor and ESP32 microcontroller" by A. Ghosh and R. Kumar, 2019 3rd International Conference on Inventive Systems and Control (ICISC).

https://ieeexplore.ieee.org/abstract/document/9035843

"Wireless ECG Monitoring System Based on the AD8232 and ESP32 Microcontroller" by J. A. Santacoloma, R. C. Orozco, and D. E. Vargas, 2020 IEEE XXVII International Conference on Electronics, Electrical Engineering and Computing (INTERCON).

https://ieeexplore.ieee.org/abstract/document/9292245