SHIV NADAR UNIVERSITY CHENNAI

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF ENGINEERING

PROJECT REPORT

BIOMEDICAL MONITORING SYSTEM

A PROJECT REPORT

Submitted by

Srividhya Asuvathraman – 23011102100

Sharon Grace Prabhu – 23011102093

Shirley Claire S – 23011102094

S Haripriya - 23011102099

Introduction to Internet of Things and Laboratory

SHIV NADAR UNIVERSITY CHENNAI

MAY 2024

BIOMEDICAL MONITORING SYSTEM

Abstract— Due to the COVID-19 pandemic, healthcare technology has gained significant prominence. This surge is fueled by advancements in Internet of Things (IoT) technologies, which are being rapidly developed and adopted. Due to busy lifestyles and daily responsibilities, continuously monitoring patient health can be challenging. This is particularly crucial for elderly patients who require regular supervision. To address this, a novel system has been developed to automate the monitoring of patient vitals, thereby enhancing the efficiency of healthcare delivery. The proposed system utilizes an array of sensors integrated with a microcontroller to continuously track and record vital signs such as body temperature, heart rate, blood oxygen levels, and ambient humidity. This paper describes the design, implementation, and functionality of this innovative patient monitoring device, highlighting its potential to transform patient care by leveraging IoT capabilities.

Keywords—IoT, ESP-32, Blood Oxygen Level, Pulse Oximeter, Microcontroller

INTRODUCTION:

OVERVIEW:

Remote monitoring of heart prefers a portable medical system by which a patient's health condition can be checked-up and the report can be visualized or monitored by a medical specialist or doctor. The report is transmitted to the medical specialist by such an electronic device which is internet enabled. The device can be laptop, mobile phone or smart watch . While talking about internet, it has become the most attached thing to the human life nowadays. Remote monitoring is a blessing of internet of things (IoT). The IoT is such a phenomenon by which an object is connected to the internet and the connection is used to control those objects or to monitor the system remotely. Sensors, Connectivity, Data processing and User interface are main components of IoT. Thus, remote monitoring is consisted with all of these combined components. Remote monitoring of healthcare system is a new idea which enables doctor or medical specialist to monitor patient's health parameters from anywhere at any time through internet enabled medical devices. Notification can be sent to the doctor if any abnormal condition is detected in patient's body. The detection is done by using some sensors or devices like pulse sensor, temperature sensor etc.

The heart is a vital organ that pumps blood between the rib cage and the lungs. It is divided into two sides, with the lower chamber called ventricles and the upper chamber called atrium. Blood is pumped through the heart through a one-way valve, with the right ventricles contracting to push blood to the lungs, and the oxygenated blood returning to the left atrium through four pulmonary veins. The atrium pumps blood back to the left ventricles, pumping all blood back to the body. The heart beats rhythmically, driven by electrical impulses generated by the conductive system of the heart. The sinoatrial node, also known as the natural pacemaker, generates these impulses, which travel through the heart in a defined rhythmic pattern. The AV node, or atrioventricular node, is the natural gatekeeper of the heart, passing the electrical impulses to the ventricles. The heart is a complex organ, and monitoring heart rate is crucial for a healthy life.

The goal of this project is to monitor and measure a range of indicators in hospitals, including pulse rate, and temperature. An ESP 32 can be used to record the results, which will then be communicated via Wi-Fi to the user-end application and shown on a monitor. Physicians may use a website or an app to review the results.

OBJECTIVES:

In this paper, the main component of the system is the microcontroller which is named ESP32. The microcontroller collects the sensor data and send them to the cloud server for remote monitoring. As ESP32 has the Wi-Fi capability thus it uses the internet connection to send the sensor data. Two sensors are used in the system. A Pulse Oximeter Sensor integrates pulse oxymetry and a heart rate monitor. MAX30100 is the sensor. It detects pulse and heart rate signals with two LEDs, a photo-detector, an optical calibration, and analog low-noise signals. Two temperature sensors, this version of the DS18B20 sensor is pre-wired and waterproofed. This tool is handy if you have to weigh anything from a distance or in damp weather. Temperatures between -55 to 125°C can be detected by the sensor. The DHT11 sensor is made up of a capacitive module for sensing humidity and a temperature sensing thermistor

The data is showed in a monitor of a personal computer/mobile phone.

1. A system that guarantees constant data transmission needs to be established in order to monitor healthcare in real time. This research proposes an efficient system with innovative sensors that can transfer data for monitoring in real time.
2. One can avoid being critically ill or finding themselves in the middle of a worldwide pandemic by using statistical analysis of real-time patient data.
3. Creating an inexpensive, effective, and dependable Internet of Things platform for the healthcare sector.

MOTIVATION:

Finding innovative and cutting-edge methods to provide the ageing population with high-quality healthcare is essential given the sharp rise in the number of senior people over the past few decades. It makes sense to treat patients with chronic illnesses or the elderly widely in order to transition healthcare facilities from hospital settings to more remote and home-based settings.
Hospitals in remote areas rarely have adequate medical facilities. Issues within the healthcare system are a result of poor health management. It ought to be beneficial to each as well. A new report by The India Spend data analysis[1] claims that there is a 500,000 doctor deficit in India. India has failed to meet the WHO's recommended doctor-to-patient ratio of 1:1000.

LITERATURE SURVEY

In recent years, there has been a surge in popularity for remote health monitoring systems, particularly those focusing on heart health. As a result, numerous researchers and authors have published works on remote heart monitoring systems. These works often involve modifications and advancements to existing systems. They include studies on real-time monitoring of patients' heart rates and other physiological parameters.

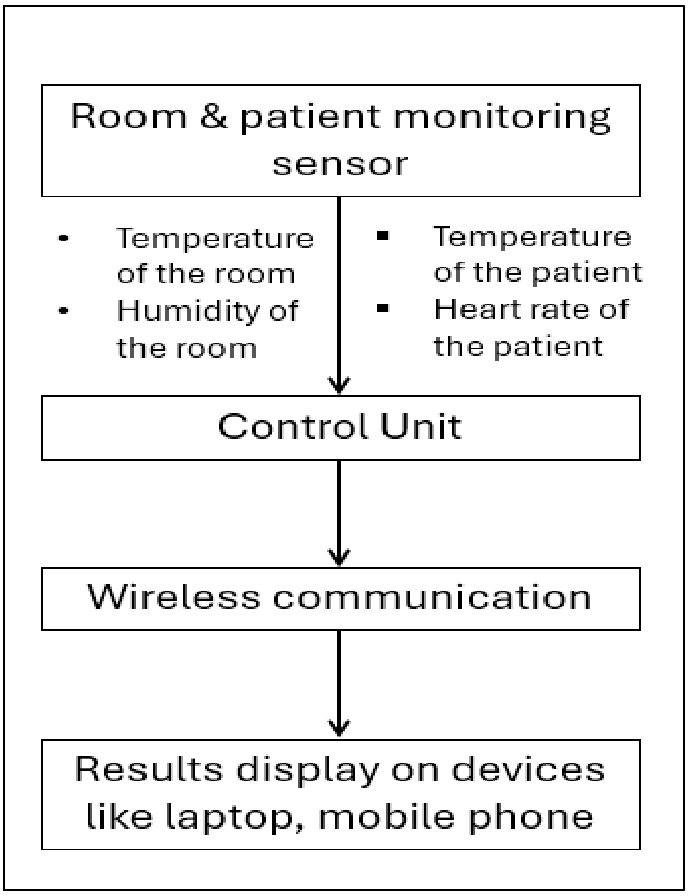
Alexander Maier, Andrew Sharp and Yuriy Vagapov (Computing and Engineering, Glyndwr University, Wrexham, UK) [1] have presented a detailed comparison of the ESP32 to its competitors including competitive analysis of its technical features and functions in their paper ‘Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things’. ESP32 is cost-effective, its dual-core processor provides a good balance of performance and power efficiency

S. Dey and T. Bera have written about innovative design and development of a smart device with IoT capabilities based on the ESP32 microcontroller in their paper "Design and Development of a Smart and Multipurpose IoT Embedded System Device Using ESP32 Microcontroller."[2] This work involves a compact embedded kit that measures atmospheric pressure, temperature, and humidity, and transmits the data to the IoT website. It features a heart rate sensor that sends data to a custom-designed Android app via Bluetooth and a compass that can sense the direction, displaying the current date and time. It can also be reprogrammed as a health monitoring system, drone controller, car controller, or different system controller like home automation, depending on the user's needs.

S. M. G. Mostafa, M. Zaki, M. M. Islam, M. S. Alam and M. A. Ullah [3]have presented a paper on an up-to-date IoT-based project that continuously monitors the patient's body temperature, heart-rate and oxygen saturation level; keep the data readings in display before the patient and in the screen of the doctor's mobile in their paper "Design and Implementation of an IoT-Based Healthcare Monitoring System." Their work design integrates NodeMCU, DS18B20 Temperature sensor, Max30100 Pulse-oximeter, and other required materials in a small box.

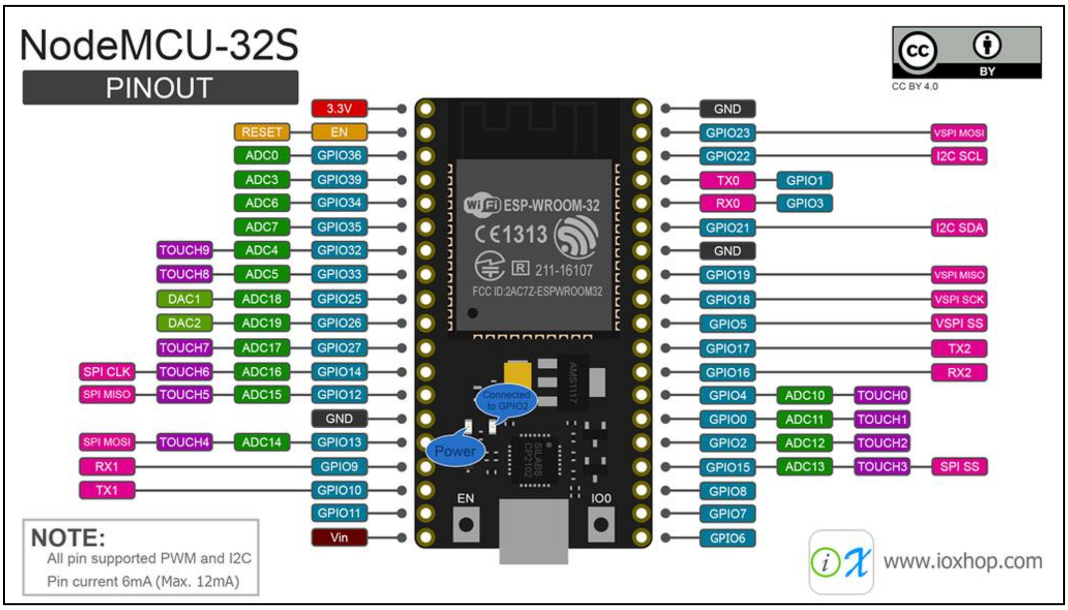
PROPOSED METHODOLOGY

BLOCK DIAGRAM:

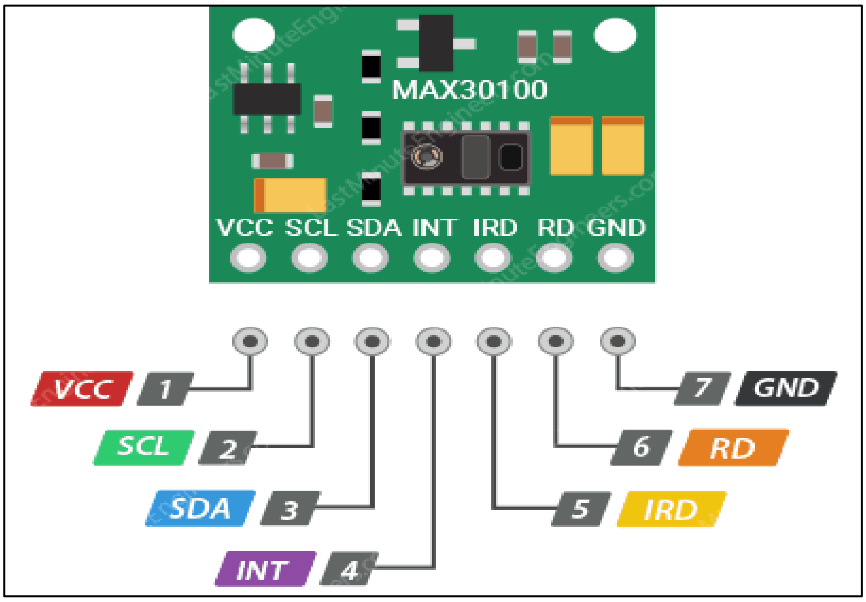


PIN DIAGRAMS:

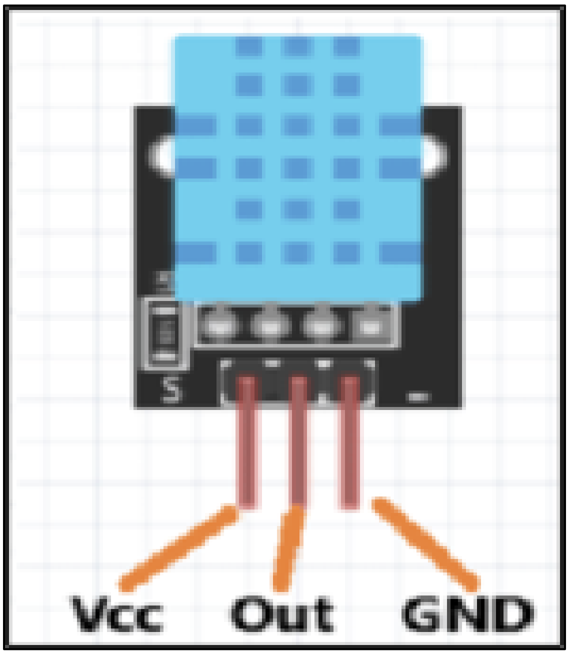
1)ESP 32



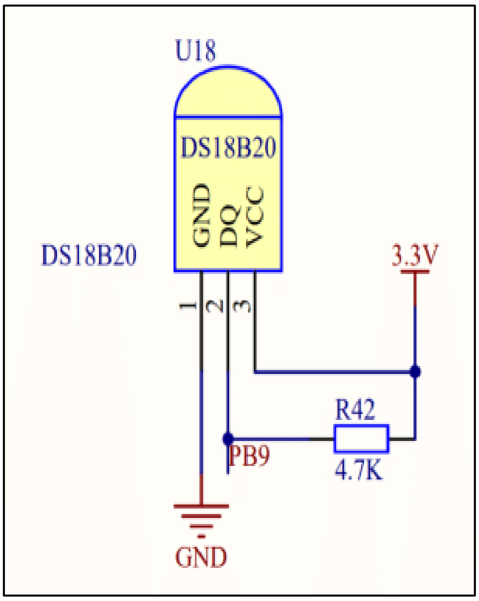
2) MAX30100



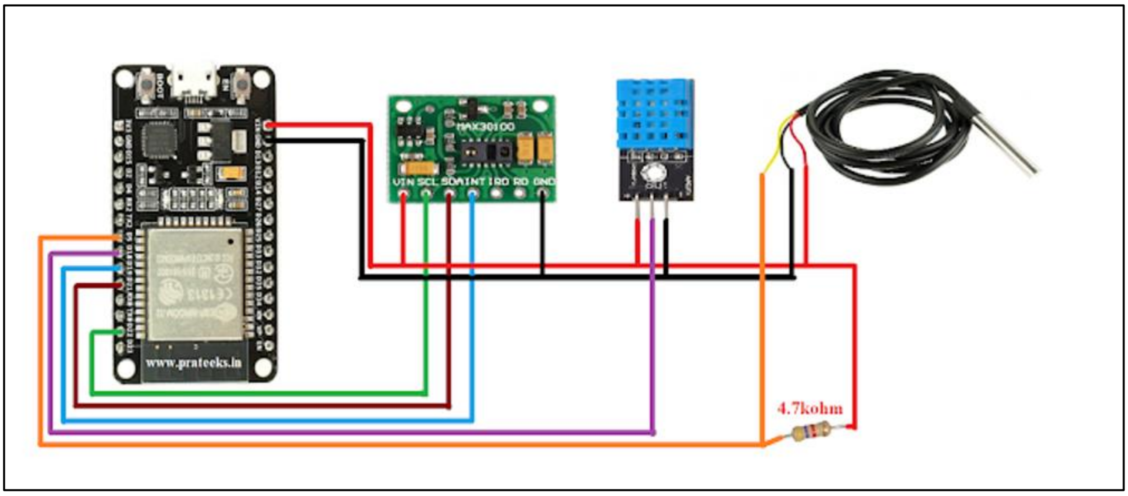
3) DHT 11



4) DS18B20



CIRCUIT DIAGRAM:



HARDWARE AND SOFTWARE USED

1)Pulse Oximeter Sensor – MAX30100

A Pulse Oximeter Sensor integrates pulse oximetry and a heart rate monitor. MAX30100/MAX30102 is the sensor. The MAX30100 uses two LEDs, an IR LED and a red LED, to measure blood oxygen saturation (SpO2) and heart rate. The sensor works based on the principle of photoplethysmography (PPG). This technique involves shining light (from the LEDs) through a part of the body, typically a fingertip or an earlobe, and measuring the light absorption with photodetectors positioned on the other side of the tissue. It is powered by 1.8V and 3.3V and can be turned off via software with very low standby power, which ensures that the power supply is always connected.

Technical specs:
Input power: 1.7 to 2.0 V
Temperature range: -40 to +85 °C
LED Current: 0mA to 50mA (typ)
LED pulse width: 200µs to 1.6ms
Supply current in shutdown: 0.7-10µA
Package: 5.6mm x 2.8mm x 1.2mm 14-Pin SiP

2)Temperature Sensor – DS18B20

This version of the DS18B20 sensor is pre-wired and waterproofed. This tool is handy if you have to weigh anything from a distance or in damp weather. Temperatures between -55 to 125°C can be detected by the sensor. The wire is jacked with PVC.

This DS18B20 temperature sensor works in the same way as a temperature sensor. The sensor's resolution varies between 9 and 12 bits. However, the default resolution for powering up is 12-bit. This sensor receives power while it is in a low power inactive state. A convert-T command can be used to calculate temperature as well as convert Ato-D values. This resulting information of temperature is stored in the sensor's 2-byte register, and the sensor then goes back to being inactive.

Technical specs:
Usable temperature range: -55 to 125°C (-67°F to +257°F)
9 to 12 bit selectable resolution
Uses 1-Wire interface- requires only one digital pin for communication
Unique 64 bit ID burned into chip
Multiple sensors can share one pin
±0.5°C Accuracy from -10°C to +85°C
Temperature-limit alarm system
Query time is less than 750ms
Usable with 3.0V to 5.5V power/data

3) Humidity & Temperature Sensor – DHT11

The DHT11 sensor is made up of a capacitive module for sensing humidity and a temperature sensing thermistor. The dielectric moisture-holding substratum serves between the two humidity sensing condenser electrodes. The capacity value changes with changes in moisture levels. The IC measures, processes and transforms the values of resistance into digital form. It uses a negative thermistor coefficient for measuring the temperature, which reduces the value of the resistance as the temperature increases. This sensor is usually made of ceramics or polymers for semiconductors to achieve a higher resistance even when the temperature is the slightest.

Technical specs:
Operating Voltage: 3.5V to 5.5V
Operating current: 0.3mA (measuring) 60uA (standby)
Output: Serial data
Temperature Range: 0°C to 50°C
Humidity Range: 20% to 90%
Resolution: Temperature and Humidity both are 16-bit
Accuracy: ±1°C and ±1%

4) ESP32 processor:

ESP32 is one of the key IoT devices for learning. It gives a full Linux framework to an exceptionally ease on a little stage. The framework sensors and actuators are fitted with GPIO pins on the ESP32. ESP32 and IoT structure a progressive innovation for advancement in medical services. The ESP32 is amazingly very much planned with its coordinated receiving wire switches, control enhancement, commotion upsetting speakers, channels and force the executives' modules. It can work alone or a host MCU slave, diminishing time went through with the principle application processor. It very well may be an independent framework. The SPI/SDIO and I2C/UART on the EPS32 permit you to speak with other Bluetooth and Wi-Fi gadgets.

This board has 2.4 GHz dual-mode Wifi and a BT wireless connection. In addition, a 512 KB SRAM and a 4 MB memory are integrated on the microcontroller development board. The board has 21 pins for interface connection, including I²C, SPI, UART, DAC and ADC.

Technical specs:
Model: NodeMCU ESP32
Type: ESP32
Processor: Tensilica LX6 Dual-Core
Clock Frequency:240 MHz
SRAM: 512 kB
Memory: 4 MB
Wireless Standard: 802.11 b/g/n
Frequency:2.4 GHz
Bluetooth Wireless Connection: Classic / LE
Data Interfaces:UART / I2C / SPI / DAC / ADC
Operating Voltage:3.3V (can be operated via 5V micro-USB)
Max. Current Draw per GPIO: 40 mA
Operating Temperature: -40°C to 125°C
Dimensions: 48mm x 26mm x 11.5mm

5)Arduino IDE

The Arduino open-source software, which is also called Arduino IDE, makes it easy to write, compile and download code. The environment is written in Java and based on software like Processing, open-source. This program is compatible with any Arduino board. A text editor for writing code, a message field, a text console, a toolbar with buttons for common functions, and a set of menus are all included in the Arduino IDE. It attaches to any hardware and allows you to upload and interact with programs. The sensors' libraries and the ESP-32 Dev Board's libraries are also installed.

METHODOLOGY

The proposed system working can be divided into three different segments. Sensor Module, Data Processing Module, WebUser Interface are the three steps in this model. The DSB180 is a temperature sensor that can supply 9 bits to 12 bits. The bit value changes depending upon the temperature. This value is fed to ESP32 through a 1-wire protocol. The DHT 11 is a humidity and temperature sensor. It calculates humidity by measuring the electrical resistance between two electrodes/These values are sent to ESP32 via I2C. MAX30102 is a pulse oximeter sensor. It can provide values of Pulse (bpm) and Oxygen level (%). It works by using red and infrared light to differentiate the absorption characteristics of oxygenated and deoxygenated haemoglobin. ESP32 collects all the data from sensors and converts them to digital form ESP32 provides data to the client through Wi-Fi. It creates a unique IP Address and contains code related to Web Page HTML. This IP Address is entered in the web browser to know the vitals at every instant. The vitals of the patient can be viewed in a web browser.

A combination of hardware components is used to implement the device. During the implementation process, all of the hardware components are put together. Figure 6 shows the circuit diagram for the built device. Since it includes a built-in Wi-Fi module, the ESP32 is used as a processing unit. The VCC and GND pins of the ESP32 are connected to the VCC and GND pins of all sensors. At 3.3V VCC, all of the sensors will operate. So make sure their VCC is connected to a 3.3V power supply. Because the MAX30100 is an I2C sensor, connect its SDA and SCL pins to GPIO21 and GPIO22, respectively. Connect its INT pin to ESP32's GPIO19. DHT11's output pin is connected to ESP32's GPIO18. Similarly, the DS18B20's output pin is connected to ESP32's GPIO5. Between the output pin and the VCC pin of the DS18B20 is a 4.7K pull-up resistor.

ESP32 SKETCH FOR BIOMEDICAL MONITORING SYSTEM:

```
#include <WiFi.h>
#include <WebServer.h>
#include <Wire.h>
#include "MAX30100_PulseOximeter.h"
#include <OneWire.h>
#include <DallasTemperature.h>
#include "DHT.h"

#define DHTTYPE DHT11
#define DHTPIN 18
#define DS18B20 5
#define REPORTING_PERIOD_MS     1000

float temperature, humidity, BPM, SpO2, bodytemperature;

/*Put your SSID & Password*/
const char* ssid = "SSN";  // Enter SSID here
const char* password = "Ssn1!Som2@Sase3#";  //Enter Password here

DHT dht(DHTPIN, DHTTYPE);;
PulseOximeter pox;
uint32_t tsLastReport = 0;
OneWire oneWire(DS18B20);
DallasTemperature sensors(&oneWire);


WebServer server(80);

void onBeatDetected()
{
  Serial.println("Beat!");
}

void setup() {
  Serial.begin(115200);
  pinMode(19, OUTPUT);
  delay(100);
  Serial.println(F("DHTxx test!"));
  dht.begin();
  Serial.println("Connecting to ");
  Serial.println(ssid);

  //connect to your local wi-fi network
  WiFi.begin(ssid, password);

  //check wi-fi is connected to wi-fi network
  while (WiFi.status() != WL_CONNECTED) {
  delay(1000);
  Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected..!");
  Serial.print("Got IP: ");  Serial.println(WiFi.localIP());

  server.on("/", handle_OnConnect);
  server.onNotFound(handle_NotFound);

  server.begin();
  Serial.println("HTTP server started");
```

```
  Serial.print("Initializing pulse oximeter..");

  if (!pox.begin()) {
    Serial.println("FAILED");
    for (;;);
  } else {
    Serial.println("SUCCESS");
    pox.setOnBeatDetectedCallback(onBeatDetected);
  }

   pox.setIRLedCurrent(MAX30100_LED_CURR_7_6MA);

  // Register a callback for the beat detection

}
void loop() {
  server.handleClient();
  pox.update();
  sensors.requestTemperatures();
  float t = dht.readTemperature();
  String Temperature_Value = String(t);
  float h = dht.readHumidity();
  String Humidity_Value = String(h);
  temperature = t;
  humidity = h;
  BPM = pox.getHeartRate();
  SpO2 = pox.getSpO2();
  bodytemperature = sensors.getTempCByIndex(0);


  if (millis() - tsLastReport > REPORTING_PERIOD_MS)
  {
    Serial.print("Room Temperature: ");
    Serial.print(t);
    Serial.println("°C");

    Serial.print("Room Humidity: ");
    Serial.print(h);
    Serial.println("%");

    Serial.print("BPM: ");
    Serial.println(BPM);

    Serial.print("SpO2: ");
    Serial.print(SpO2);
    Serial.println("%");

    Serial.print("Body Temperature: ");
    Serial.print(bodytemperature);
    Serial.println("°C");

    Serial.println("*********************************");
    Serial.println();

    tsLastReport = millis();
  }

}

void handle_OnConnect() {
```

```
  server.send(200, "text/html", SendHTML(temperature, humidity, BPM, SpO2, bodytemperature));
}

void handle_NotFound(){
  server.send(404, "text/plain", "Not found");
}

  String SendHTML(float temperature, float humidity, float BPM, float SpO2, float bodytemperature) {
  String html = "<!DOCTYPE html>";
html += "<html>";
html += "<head>";
html += "<title>Patient Health Monitoring</title>";
html += "<meta name='viewport' content='width=device-width, initial-scale=1.0'>";
html += "<link rel='stylesheet' href='https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.7.2/css/all.min.css'>";
html += "<link rel='stylesheet' type='text/css' href='styles.css'>";
html += "<style>";
html += "body { background-color: #fff; font-family: sans-serif; color: #333333; font: 14px Helvetica, sans-serif box-sizing: border-box;}";
html += "#page { margin: 20px; background-color: #fff;}";
html += ".container { height: inherit; padding-bottom: 20px;}";
html += ".header { padding: 20px;}";
html += ".header h1 { padding-bottom: 0.3em; color: #008080; font-size: 45px; font-weight: bold; font-family: Garmond, 'sans-serif'; text-align: center;}";
html += "h2 { padding-bottom: 0.2em; border-bottom: 1px solid #eee; margin: 2px; text-align: left;}";
html += ".header h3 { font-weight: bold; font-family: Arial, 'sans-serif'; font-size: 17px; color: #b6b6b6; text-align: center;}";
html += ".box-full { padding: 20px; border 1px solid #ddd; border-radius: 1em 1em 1em 1em; box-shadow: 1px 7px 7px 1px rgba(0,0,0,0.4); background: #fff; margin: 20px; width: 300px;}";
html += "@media (max-width: 494px) { #page { width: inherit; margin: 5px auto; } #content { padding: 1px;} .box-full { margin: 8px 8px 12px 8px; padding: 10px; width: inherit;; float: none; } }";
html += "@media (min-width: 494px) and (max-width: 980px) { #page { width: 465px; margin 0 auto; } .box-full { width: 380px; } }";
html += "@media (min-width: 980px) { #page { width: 930px; margin: auto; } }";
html += ".sensor { margin: 12px 0px; font-size: 2.5rem;}";
html += ".sensor-labels { font-size: 1rem; vertical-align: middle; padding-bottom: 15px;}";
html += ".units { font-size: 1.2rem;}";
html += "hr { height: 1px; color: #eee; background-color: #eee; border: none;}";
html += "</style>";

//Ajax Code Start
  html += "<script>\n";
  html += "setInterval(loadDoc,1000);\n";
  html += "function loadDoc() {\n";
  html += "var xhttp = new XMLHttpRequest();\n";
  html += "xhttp.onreadystatechange = function() {\n";
  html += "if (this.readyState == 4 && this.status == 200) {\n";
  html += "document.body.innerHTML =this.responseText}\n";
  html += "};\n";
```

```
  html += "xhttp.open(\"GET\", \"/\", true);\n";
  html += "xhttp.send();\n";
  html += "}\n";
  html += "</script>\n";
  //Ajax Code END

html += "</head>";
html += "<body>";
html += "<div id='page'>";
html += "<div class='header'>";
html += "<h1>Biomedical Monitoring System</h1>";
html += "</div>";
html += "<div id='content' align='center'>";
html += "<div class='box-full' align='left'>";
html += "<h2>Sensors Readings</h2>";
html += "<div class='sensors-container'>";

//For Temperature
html += "<div class='sensors'>";
html += "<p class='sensor'>";
html += "<i class='fas fa-thermometer-half' style='color:#0275d8'></i>";
html += "<span class='sensor-labels'> Room Temperature </span>";
html += (int)temperature;
html += "<sup class='units'>°C</sup>";
html += "</p>";
html += "<hr>";
html += "</div>";

//For Humidity
html += "<div class='sensors'>";
html += "<p class='sensor'>";
html += "<i class='fas fa-tint' style='color:#5bc0de'></i>";
html += "<span class='sensor-labels'> Room Humidity </span>";
html += (int)humidity;
html += "<sup class='units'>%</sup>";
html += "</p>";
html += "<hr>";

//For Heart Rate
html += "<p class='sensor'>";
html += "<i class='fas fa-heartbeat' style='color:#cc3300'></i>";
html += "<span class='sensor-labels'> Heart Rate </span>";
html += (int)BPM;
html += "<sup class='units'>BPM</sup>";
html += "</p>";
html += "<hr>";

//For Sp02
html += "<p class='sensor'>";
html += "<i class='fas fa-burn' style='color:#f7347a'></i>";
html += "<span class='sensor-labels'> Sp02 </span>";
html += (int)SpO2;
html += "<sup class='units'>%</sup>";
html += "</p>";
html += "<hr>";

//For Body Temperature
html += "<p class='sensor'>";
html += "<i class='fas fa-thermometer-full' style='color:#d9534f'></i>";
html += "<span class='sensor-labels'> Body Temperature </span>";
html += (int)bodytemperature;
```

```
html += "<sup class='units'>°C</sup>";
html += "</p>";
html += "</div>";
html += "</div>";
html += "</div>";
html += "</div>";
if (BPM > 60) {
html += "<p id='more_beat' style='color:red;font-size:24px;font-weight:800' align='center'><i class='fas fa-triangle-exclamation'></i>The heart beat is more than 60 BPM!</p>";
}
html += "</div>";
html += "</body>";
html += "</html>";
return html;
}
```

APPLICATIONS:

"Health Monitoring System Based on IoT," the proposed system has various applications in the healthcare industry and beyond. Here are some potential applications of this health monitoring system based on IoT technology:

1. Remote Patient Monitoring: The system allows for continuous monitoring of vital signs such as body temperature, heart rate, blood oxygen levels, and humidity remotely. This is particularly useful for patients with chronic conditions or the elderly who require regular monitoring without the need for frequent hospital visits.
2. Early Disease Detection: By continuously monitoring vital signs, the system can help in early detection of health issues or abnormalities. This proactive approach can lead to timely interventions and better health outcomes for patients.
3. Telemedicine: The data collected by the health monitoring system can be transmitted to healthcare providers in real-time, enabling telemedicine consultations. This can improve access to healthcare services, especially in remote areas where medical facilities are limited.
4. Health and Fitness Tracking: The system can also be used for personal health and fitness tracking. Individuals can monitor their vital signs and track their progress towards fitness goals, leading to a healthier lifestyle.
5. Emergency Response: In case of emergencies, the system can alert healthcare providers or emergency services automatically based on predefined thresholds or abnormal readings. This can ensure timely assistance in critical situations.
6. Clinical Research: The data collected by the system can be used for clinical research purposes, providing valuable insights into population health trends, disease patterns, and treatment outcomes.
7. Elderly Care: The system is particularly beneficial for elderly care, where continuous monitoring of vital signs can help in ensuring the well-being of senior citizens living alone or in assisted living facilities.
8. Hospital Management: In hospital settings, the system can streamline patient monitoring processes, improve efficiency in healthcare delivery, and enhance overall patient care.

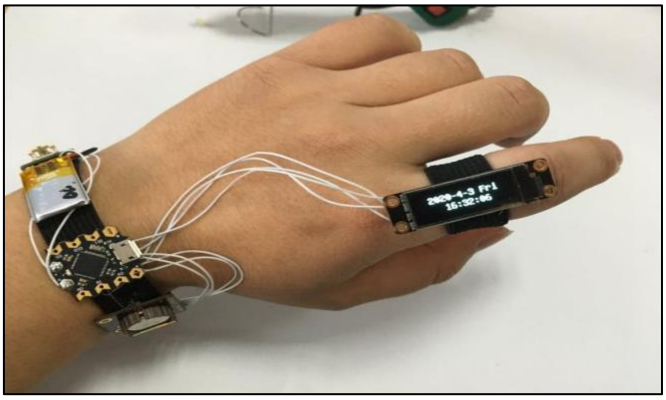
Overall, the health monitoring system based on IoT technology offers a wide range of applications that can revolutionize the healthcare industry by providing personalized, efficient, and proactive healthcare solutions.

INFERENCE OF MINI PROJECT:

[4]Title: IoT-Based Smart Health Monitoring Bracelet

Objective: Develop a wearable smart health monitoring bracelet to track vital signs and provide real-time data to users.

- Components: Sensors, Microcontroller, Bluetooth Module, Battery, LED Display.
- Features: Continuous Monitoring, Mobile App Integration, Health Insights, Alert System, Data Storage.
- Implementation Steps: Sensor Integration, Microcontroller Programming, Mobile App Development, User Interface Design, Testing.
- Conclusion: The IoT-based smart health monitoring bracelet project offers a practical application of wearable technology and IoT in personal health tracking, empowering users with continuous monitoring and data-driven insights



QUERIES:

1)What is PPG?

Photoplethysmography (PPG) is a simple optical technique used to detect volumetric changes in blood in peripheral circulation. It is a low cost and non-invasive method that makes measurements at the surface of the skin. The technique provides valuable information related to our cardiovascular system.

-How Does It Work in pulse oximeter?

Photoplethysmography (PPG) works by shining light into the skin and tissues and then detecting the amount of light that is either transmitted through or reflected back. The changes in blood flow, particularly in the arteries and arterioles, cause fluctuations in the amount of light detected. These fluctuations are mainly due to the pulsatile nature of blood flow during the cardiac cycle.

Light sources used in PPG typically include red and infrared (IR) light, as they penetrate tissue and blood effectively. Green light is also used but to a lesser extent. LEDs have become more efficient over time, leading to better PPG performance.

PPG sensors can operate in two modes: transmission and reflectance. In transmission mode, the light transmitted through the tissue is detected, while in reflectance mode, the light reflected or backscattered from tissue is detected. These modes provide flexibility in sensor placement and usage.

2)Can ESP32 run on battery?

Yes ESP32 can run on battery.

Power Supply for NodeMCU by Battery:

- The NodeMCU can be powered by a 3.7V Lithium-Ion battery.

- However, it's not safe to directly power the NodeMCU from a lithium-ion cell because its nominal voltage is usually at 3.7V or higher, exceeding the NodeMCU's maximum input voltage of 3.6V. This could potentially damage the board.(either we can use boost converter or dc adaptor).

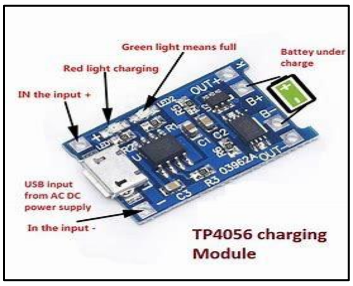
- Boost Converter Module: Instead of a DC adapter, you can use a 3.7V lithium-ion or lithium-polymer battery along with a boost converter module. This module can raise the voltage from 3.7V to 5V, which is suitable for the NodeMCU.

- Components needed for this method:

  - 3.7V lithium-ion battery

  - Boost converter module

  - TP4056 Battery Charger



- Procedure:

  - Use a 3.7V Lithium-Ion or Lithium Polymer battery.

  - Utilize a Boost Converter Module to raise the voltage from 3.7V to 5V.

  - Connect the Boost Converter's output to the 5V VIN pin of the NodeMCU.

  - Connect the battery terminal to the output terminal of the TP4056 Battery Charger Module for charging.

3)Can code be loaded onto ESP32?

ESP32 is a development board that includes an ESP32 microcontroller, which is a low-power, low-cost microcontroller with built-in WiFi and Bluetooth capabilities. The development board also includes various other components such as USB ports, power supply, and a programming interface that allow you to upload code to the ESP32.So yes, code can be loaded onto ESP32.

4)Is ESP32 capable of wireless communication?

The ESP32 is renowned for its robust wireless communication capabilities, making it ideal for diverse IoT, home automation, and wireless control applications. Here's a summary of its wireless features:

1. Wi-Fi Connectivity: The ESP32 supports Wi-Fi, enabling it to connect to networks as a client or create its own access point. This facilitates data exchange, remote control, and internet connectivity.

2. Bluetooth Support: It features Bluetooth Classic and Bluetooth Low Energy (BLE) capabilities, allowing seamless communication with smartphones, wearables, and other devices for various applications.

3. Bluetooth Mesh: ESP32 supports Bluetooth Mesh networking, enabling decentralized communication among multiple devices, useful for large-scale IoT deployments and smart systems.

4. Dual-mode Operation: The ESP32 can operate concurrently in Wi-Fi and Bluetooth modes, expanding its functionality and application scope.

5. Other Wireless Protocols: Through software-defined radio (SDR), the ESP32 can accommodate additional protocols like Zigbee, LoRa, and RF, enhancing its versatility and compatibility with various systems.

CONCLUSION AND FUTURE WORKS:

The report presents a prototype for a health monitoring system based on IoT, aimed at continuously tracking various health parameters and predicting potential illnesses or conditions. It utilizes sensors to collect data such as body temperature, heart rate, and room humidity, which are then processed by an ESP32 microcontroller and transmitted via Wi-Fi to a web user interface. The system can be deployed in hospitals to facilitate remote monitoring and storage of patient data in an online database.

As for future work, enhancing the device by incorporating artificial intelligence elements to support healthcare professionals and patients. They propose utilizing data mining and GPS monitoring to identify consistent trends and hierarchical relationships in disease patterns, enabling the prediction of potential health issues based on historical data. This approach could aid in diagnosing and treating patients more effectively by leveraging insights derived from the analysis of large datasets.

REFERENCES:

[1] Maier, A. Sharp and Y. Vagapov, "Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things," 2017 Internet Technologies and Applications (ITA), Wrexham, UK, 2017, pp. 143-148, doi: 10.1109/ITECHA.2017.8101926. keywords: {Microcontrollers;Wireless fidelity;Clocks;C++ languages;Internet of Things;Bluetooth;Oscilloscopes;ESP32;Internet of Things;Wi-Fi;oscilloscope}

[2] S. Dey and T. Bera, "Design and Development of a Smart and Multipurpose IoT Embedded System Device Using ESP32 Microcontroller," 2023 International Conference on Electrical, Electronics, Communication and Computers (ELEXCOM), Roorkee, India, 2023, pp. 1-6, doi: 10.1109/ELEXCOM58812.2023.10370327. keywords: {Temperature measurement;Temperature sensors;Temperature distribution;Embedded systems;Atmospheric measurements;Microcontrollers;Control systems;IoT Devices;ESP32 microcontroller;drone controller;health monitoring;home automation;embedded systems}

[3] S. M. G. Mostafa, M. Zaki, M. M. Islam, M. S. Alam and M. A. Ullah, "Design and Implementation of an IoT-Based Healthcare Monitoring System," 2022 International Conference on Innovations in Science, Engineering and Technology (ICISET), Chittagong, Bangladesh, 2022, pp. 362-366, doi: 10.1109/ICISET54810.2022.9775850. keywords: {Temperature sensors;Temperature measurement;Heart;COVID-19;Costs;Medical services;Sensors;Max30100;DS18B20;Healthcare Monitoring;Pulse-rate;Oxygen Saturation Level;SpO2;Temperature Sensor;Automatic Hand Sanitizer}

[4] Mohammed, Bzhar & Hasan, Dler. (2023). Smart Healthcare Monitoring System Using IoT. International Journal of Interactive Mobile Technologies (iJIM). 17. 141-152. 10.3991/ijim.v17i01.34675.