Institute of Engineering and Technology (IET)

IoT based Health Monitoring System

PREPARED BY

Anay Sinhal (2021BTech012)
Chitraksh Gupta (2021BTech063)
Kanchi Sharma (2021BTech054)

FACULTY GUIDE

Dr. Hanuman Prasad Agarwal
Mr. Divanshu Jain

November, 2023

TABLE OF CONTENTS

ABSTRACT

The Internet of Things (IoT) has revolutionized the healthcare sector by enabling seamless connectivity between devices and the internet. This report details the development of an IoT-based health monitoring system using the NodeMCU ESP8266 and MAX30100 sensor for SpO2 and Heart rate measurements. In this project, we have established a two-way communication channel through the Blynk app, facilitating real-time interaction between patients and healthcare providers.

The NodeMCU ESP8266 serves as the central hub, seamlessly connecting the MAX30100 sensor for accurate SpO2 and Heart rate monitoring. The integration of Blynk app empowers patients to actively engage in their health monitoring, while healthcare professionals can remotely access vital health parameters. Additionally, the system incorporates alert mechanisms, such as a buzzer and alert lights, to notify patients promptly in critical situations.

Unlike traditional methods, our system leverages the power of the internet to transmit health data to the ThingSpeak cloud. This cloud-based approach ensures continuous and secure monitoring of health metrics. The ESP8266 Wi-Fi module facilitates the connection between the Arduino-based system and the internet, enabling global accessibility to health data.

Through this comprehensive IoT health monitoring system, we aim to enhance patient care by providing real-time data insights and timely alerts, fostering a proactive approach to healthcare management. The project not only demonstrates the potential of IoT in healthcare but also highlights the importance of remote monitoring for improved patient outcomes.

PROBLEM STATEMENT

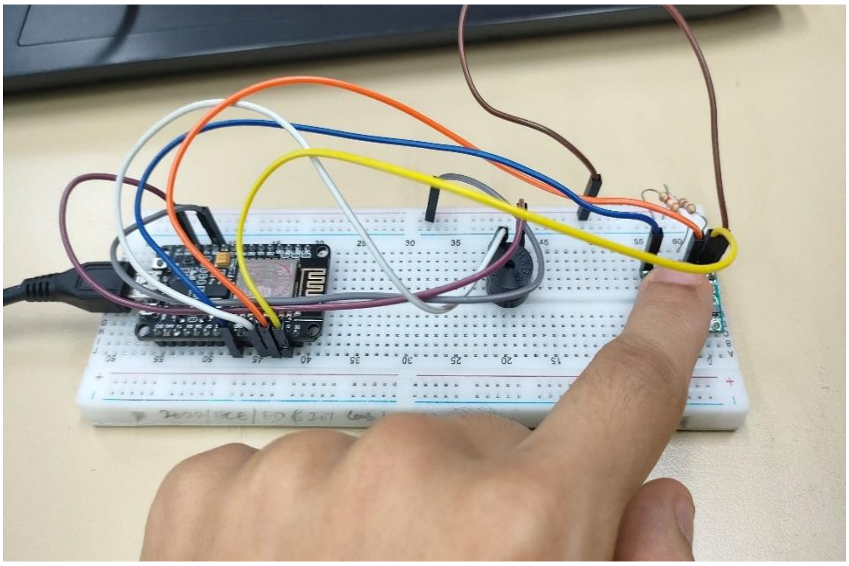
In today's fast-paced lifestyle, regular health monitoring becomes challenging due to the impracticality of frequent doctor visits. Existing systems either rely on conventional diagnostic methods or expensive smartwatches that are not specifically tailored for healthcare. These systems often lack automation, requiring human intervention, and may not cover a comprehensive set of health parameters. Additionally, the need for invasive connections for measurements poses discomfort and limits user adherence. The challenge is to create an efficient and automated health monitoring system that addresses these issues, ensuring continuous monitoring, accurate data collection, and real-time accessibility for both patients and healthcare providers.

METHODOLOGY

The development of the IoT-based health monitoring system was an iterative process involving a series of experimentation and refinement. The primary components of the system include the NodeMCU ESP8266, MAX30100 sensor for SpO2 and Heart rate measurements, and the Arduino Uno board, acting as the central processing unit. The methodology unfolded in several key stages:

Initial Sensor Integration:

We initiated the project by integrating the MAX30100 sensor for SpO2 and Heart rate measurements. This phase involved obtaining readings and displaying them within the Serial Monitor for initial data validation.

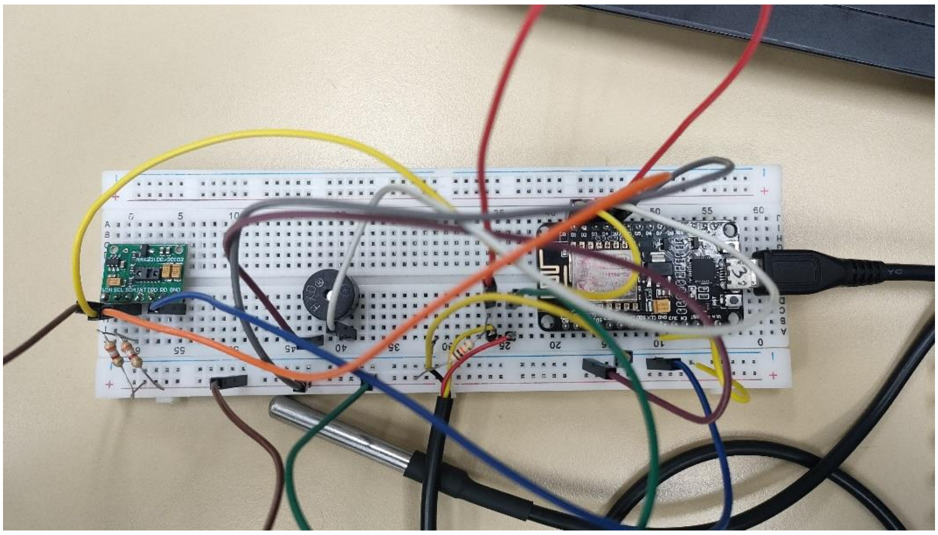


Exploration of Additional Sensors:

Recognizing the importance of comprehensive health monitoring, we experimented with additional sensors. The DS18B20 temperature sensor was considered for measuring body temperature. However, after thorough evaluation, we opted against its inclusion due to factors such as practicality, accuracy, and user comfort.

Integration of Pulse Oximeter for Blood Pressure Measurement:

To enhance the system's capabilities, we incorporated a pulse oximeter to measure blood pressure. This step aimed to provide a more holistic health profile, covering vital parameters beyond SpO2 and Heart rate.

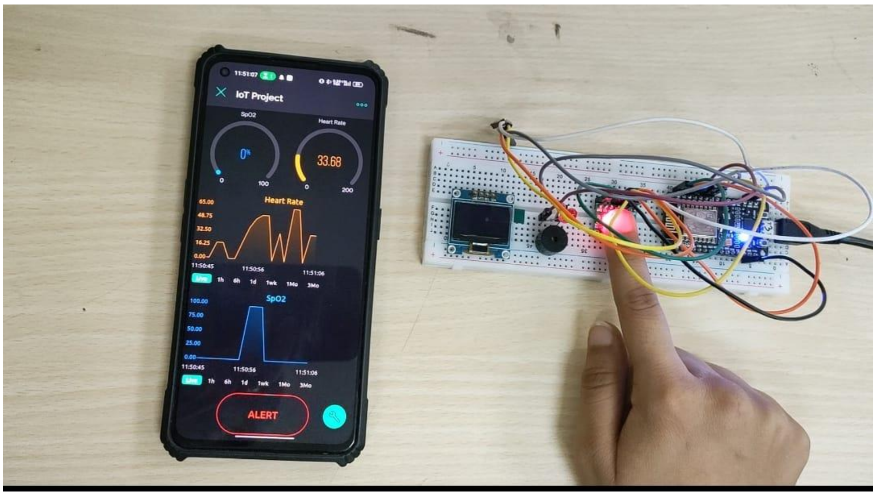


Consideration of Cloud Platforms:

Initially, we explored ThingSpeak as a cloud platform for data storage and analysis. We implemented a web-based approach, utilizing ThingSpeak to visualize and store the health data. However, after careful consideration, we found Blynk to be a more user-friendly and mobile-centric solution, facilitating real-time interaction. This decision was driven by the need for seamless mobility and accessibility, given the nature of the health monitoring system.

Implementation of Blynk App for Real-Time Interaction:

The Blynk app was selected for its intuitive mobile interface, allowing two-way communication between patients and doctors. This choice not only enhanced user experience but also provided a more convenient and accessible platform for health monitoring.



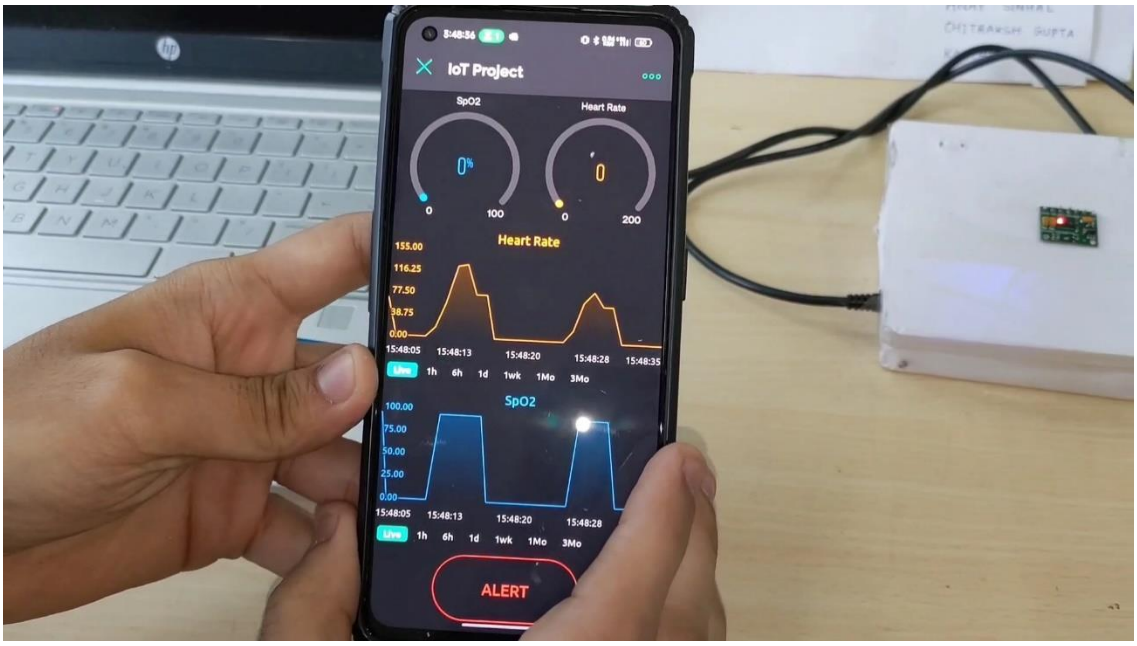
Wireless Data Transmission to ThingSpeak:

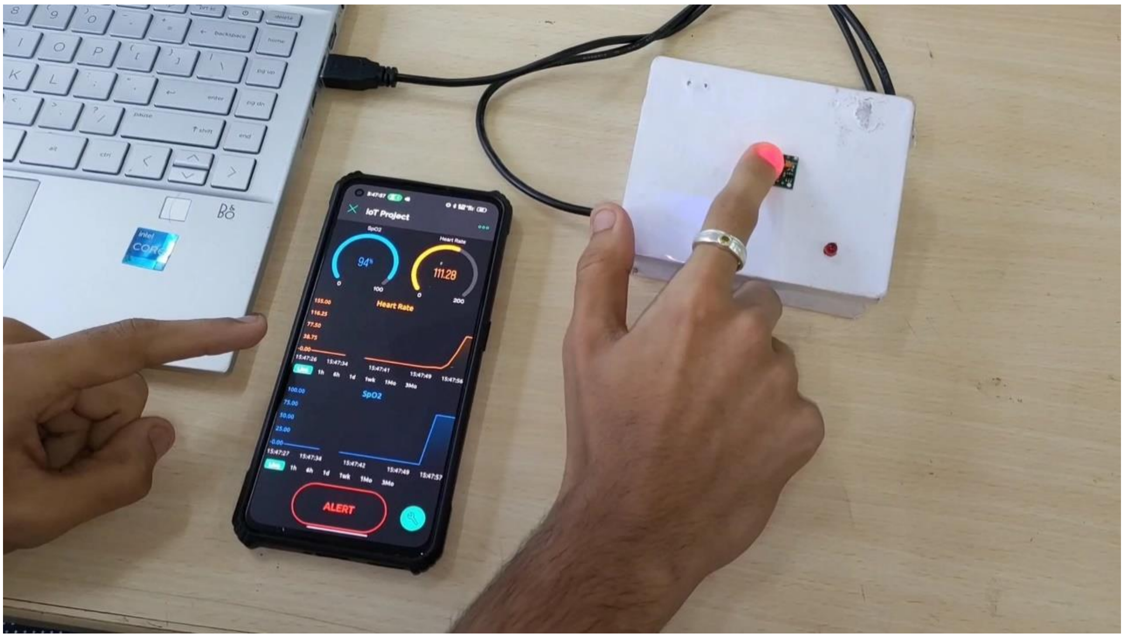
The Arduino Uno board was configured to collect, process, and wirelessly transmit the health data to the ThingSpeak cloud. This cloud-based approach ensures continuous monitoring and secure storage of health metrics, facilitating remote analysis and accessibility.

The iterative nature of the methodology allowed for the incorporation of valuable insights gained from each stage, resulting in the development of a robust IoT-based health monitoring system that prioritizes user experience, comprehensive health data, and real-time interaction.

RESULTS AND DISCUSSION

The system successfully achieves 24x7 health monitoring, providing a wide range of health parameters through non-invasive and comfortable wearable devices. The integration of advanced sensors enhances the accuracy of measurements. The wireless connectivity and IoT implementation ensure easy accessibility and real-time monitoring from any location. Results indicate the feasibility and reliability of the system in continuously tracking vital health metrics. The discussion encompasses the advantages over existing systems, emphasizing the importance of non-invasiveness, automation, and the use of IoT in healthcare.


IoT Project
SpO2
Heart Rate
Heart Rate
SpO2
ALERT


IoT Project
ALERT

HARDWARE:

1. **ESP8266:** The ESP8266 is a versatile and cost-effective Wi-Fi-enabled microcontroller, renowned for its reliability. It serves as a powerful foundation for IoT projects, providing seamless wireless connectivity and robust performance.

2. **MAX30100**: The MAX30100 is a compact and integrated pulse oximeter and heart-rate sensor module. Leveraging advanced optical sensing technology, it allows for non-invasive monitoring of blood oxygen levels and heart rate, making it valuable in health and fitness applications.

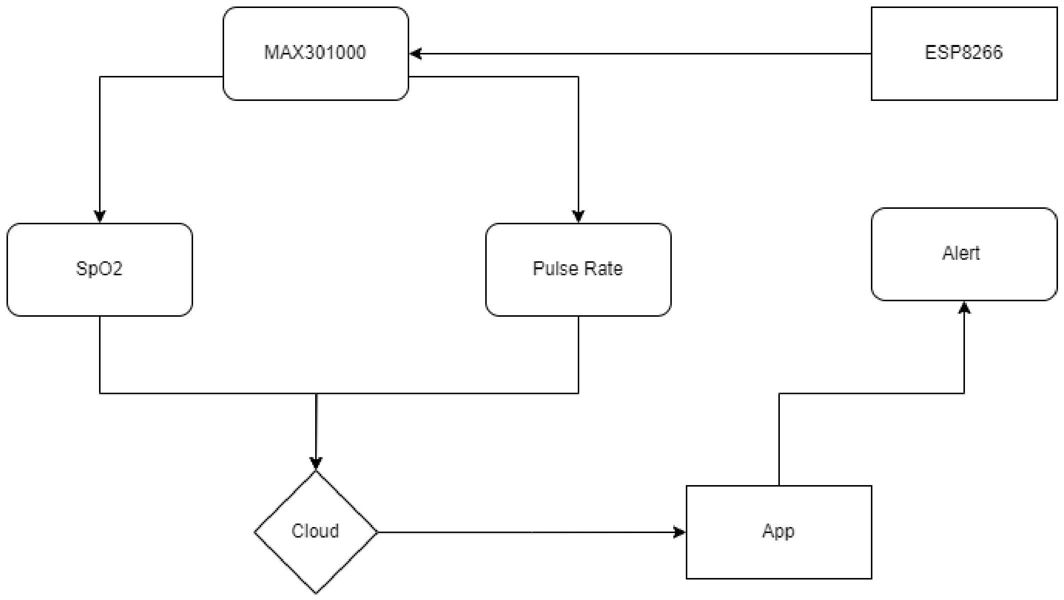
3. **Buzzer:** A buzzer is an electromechanical device that produces audible sound signals when an electric current is applied. Widely used for alert systems, notifications, or feedback in electronic projects, buzzers provide a distinctive and attention-grabbing auditory signal.

4. **LED**: Light-Emitting Diodes (LEDs) are semiconductor devices that emit light when an electric current passes through them. Commonly used for visual indicators, status notifications, or decorative lighting, LEDs come in various colors and sizes, offering energy-efficient and versatile solutions for electronic applications.

SOFTWARE:

1. Arduino
2. Blynk

HARDWARE DIAGRAM:



CONCLUSIONS

The project presents a robust solution to the challenges posed by traditional health monitoring systems. By incorporating IoT technologies and wearable devices, the system allows for continuous and automated health monitoring. The integration of advanced sensors enhances the range of parameters monitored, providing a comprehensive health profile. The two-way communication feature through the Blynk app facilitates real-time interaction between patients and healthcare providers. This approach not only addresses the limitations of existing systems but also promotes proactive healthcare management. The successful implementation and results underscore the potential of IoT in revolutionizing health monitoring for improved patient outcomes.

REFERENCES

1. A Detailed Research on Human Health Monitoring System Based on Internet of Things https://www.hindawi.com/journals/wcmc/2021/5592454/
2. IoT-Based Healthcare-Monitoring System towards Improving Quality of Life: A Review https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9601552/
3. Internet of Things (IoT): BLYNK Framework for Smart Home https://knepublishing.com/index.php/Kne-Social/article/view/4128/8495
4. Development of Blynk IoT Platform Weather Information Monitoring System: https://www.researchgate.net/publication/359177043_Development_of_Blynk_IoT_Platform_Weather_Information_Monitoring_System
5. MAX30100 Datasheet : https://www.alldatasheet.com/view.jsp?Searchword=Max30100%20datasheet&gad_source=1&gclid=Cj0KCQiAsburBhCIARIsAExmsu655s0GFPmgwzKJuRC2ZNsbkguNP41Z3a5NxoRq-d32wtxg4dgno0saAr-5EALw_wcB
6. https://www.researchgate.net/publication/340665031_IOT_BASED_HEALTH_MONITORING_SYSTEM

APPENDICES

```
#define BLYNK_TEMPLATE_ID "TMPL3o9Bpd-zB"
#define BLYNK_TEMPLATE_NAME "IoT Project"
#define BLYNK_AUTH_TOKEN "YURNrAtqysxQMzH1djcDvbHvZLWDgkbv"

#include <Wire.h>
#include "MAX30100_PulseOximeter.h"
#define BLYNK_PRINT Serial
#include <Blynk.h>
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

#include "Wire.h"
#include "Adafruit_GFX.h"
#include "OakOLED.h"

#define REPORTING_PERIOD_MS 1000
OakOLED oled;

char ssid[] = "Hotstop";  // type your wifi name
char pass[] = "abcd1234";  // type your wifi password
char auth[] = BLYNK_AUTH_TOKEN;

const int buzzerPin = D3;

// Connections : SCL PIN - D1 , SDA PIN - D2 , INT PIN - D0
PulseOximeter pox;

float BPM, SpO2;
uint32_t tsLastReport = 0;

const unsigned char bitmap [] PROGMEM =
{
  0x00, 0x00, 0x00, 0x00, 0x01, 0x80, 0x18, 0x00, 0x0f, 0xe0, 0x7f,
0x00, 0x3f, 0xf9, 0xff, 0xc0,
  0x7f, 0xf9, 0xff, 0xc0, 0x7f, 0xff, 0xff, 0xe0, 0x7f, 0xff, 0xff,
0xe0, 0xff, 0xff, 0xff, 0xf0,
  0xff, 0xf7, 0xff, 0xf0, 0xff, 0xe7, 0xff, 0xf0, 0xff, 0xe7, 0xff,
0xf0, 0x7f, 0xdb, 0xff, 0xe0,
  0x7f, 0x9b, 0xff, 0xe0, 0x00, 0x3b, 0xc0, 0x00, 0x3f, 0xf9, 0x9f,
0xc0, 0x3f, 0xfd, 0xbf, 0xc0,
```

```
  0x1f, 0xfd, 0xbf, 0x80, 0x0f, 0xfd, 0x7f, 0x00, 0x07, 0xfe, 0x7e,
0x00, 0x03, 0xfe, 0xfc, 0x00,
  0x01, 0xff, 0xf8, 0x00, 0x00, 0xff, 0xf0, 0x00, 0x00, 0x7f, 0xe0,
0x00, 0x00, 0x3f, 0xc0, 0x00,
  0x00, 0x0f, 0x00, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00
};

void onBeatDetected()
{
  Serial.println("Beat Detected!");
  oled.drawBitmap( 60, 20, bitmap, 28, 28, 1);
  oled.display();
}

void setup()
{
  Serial.begin(115200);
  oled.begin();
  oled.clearDisplay();
  oled.setTextSize(1);
  oled.setTextColor(1);
  oled.setCursor(0, 0);

  oled.println("Initializing pulse oximeter..");
  oled.display();

  pinMode(16, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
  Blynk.begin(auth, ssid, pass);

  Serial.print("Initializing Pulse Oximeter..");

  if (!pox.begin())
  {
    Serial.println("FAILED");
    oled.clearDisplay();
    oled.setTextSize(1);
    oled.setTextColor(1);
    oled.setCursor(0, 0);
    oled.println("FAILED");
    oled.display();
    for (;;);
  }
```

```
  else
  {
    oled.clearDisplay();
    oled.setTextSize(1);
    oled.setTextColor(1);
    oled.setCursor(0, 0);
    oled.println("SUCCESS");
    oled.display();
    Serial.println("SUCCESS");
    pox.setOnBeatDetectedCallback(onBeatDetected);
  }
}

void loop()
{
  pox.update();
  Blynk.run();

  BPM = pox.getHeartRate();
  SpO2 = pox.getSpO2();
  if (millis() - tsLastReport > REPORTING_PERIOD_MS)
  {
    Serial.print("Heart rate:");
    Serial.print(BPM);
    Serial.print(" bpm / SpO2:");
    Serial.print(SpO2);
    Serial.println(" %");
    if (BPM >= 120) {
      // Turn on the buzzer
      digitalWrite(buzzerPin, HIGH);
    } else {
      // Turn off the buzzer
      digitalWrite(buzzerPin, LOW);
    }

    Blynk.virtualWrite(V1, BPM);
    Blynk.virtualWrite(V2, SpO2);

    oled.clearDisplay();
    oled.setTextSize(1);
    oled.setTextColor(1);
    oled.setCursor(0, 16);
    oled.println(pox.getHeartRate());
```

```
    oled.setTextSize(1);
    oled.setTextColor(1);
    oled.setCursor(0, 0);
    oled.println("Heart BPM");

    oled.setTextSize(1);
    oled.setTextColor(1);
    oled.setCursor(0, 30);
    oled.println("Spo2");

    oled.setTextSize(1);
    oled.setTextColor(1);
    oled.setCursor(0, 45);
    oled.println(pox.getSpO2());
    oled.display();

    tsLastReport = millis();
  }
}

BLYNK_WRITE(V3)
{
  int buzzerControl = param.asInt();  // Get the state of the button
from the app
  if (buzzerControl) {
    // Turn on the buzzer
    digitalWrite(buzzerPin, HIGH);
  } else {
    // Turn off the buzzer
    digitalWrite(buzzerPin, LOW);
  }
}
```