GOVERNMENT OF TAMILNADU

DIRECTORATE OF TECHNICAL EDUCATION, CHENNAI

NAAN MUDHALVAN SCHEME (TNSDC) SPONSORED

STUDENTS DEVELOPMENT PROGRAMME

ON

IoT AND ITS APPLICATIONS

HOST INSTITUTION

xxxxx

COIMBATORE – 04

TRAINING PARTNER

ENTHU TECHNOLOGY SOLUTIONS INDIA PVT LTD

DATE:

Name	Roll No
Name 1	1
Name 2	2
Name 3	3
Name 4	4
Name 5	5

Table Of Contents

Abstract

In the modern healthcare landscape, continuous monitoring of vital health parameters is essential for ensuring timely intervention and improving patient outcomes. The IoT-Based Health Monitoring System is designed to track critical health metrics such as heart rate, blood oxygen levels, and body temperature in real-time. This project leverages the ESP32 microcontroller, which acts as the central unit for collecting, processing, and transmitting data from various sensors to the ThingzMate Cloud for remote monitoring.

The system employs sensors like the MAX30105 to measure heart rate and SpO2, and the LM35 to monitor body temperature. The collected data is displayed locally on an LCD screen and transmitted via Wi-Fi to the ThingzMate Cloud, where it can be accessed by healthcare providers or caregivers. Alerts are triggered if any of the measured parameters exceed predefined thresholds, ensuring prompt action in critical situations.

In addition to real-time monitoring, the system features an uplink for remote communication and an authorization mechanism to secure access to sensitive health data. By integrating traditional health monitoring with IoT capabilities, this project presents a scalable solution that can be easily adapted for home healthcare, remote patient monitoring, or emergency medical applications.

Introduction

In the current era of healthcare, the ability to monitor patients' vital signs continuously is crucial for ensuring their well-being and enabling timely medical intervention. Traditional health monitoring methods often require patients to be physically present in healthcare facilities, which can be impractical for those with chronic conditions or in remote areas. The IoT-Based Health Monitoring System addresses these challenges by enabling real-time tracking of critical health metrics remotely.

This system is built around the ESP32 microcontroller, which serves as the central hub for collecting data from various sensors. The MAX30105 sensor is used to measure heart rate and blood oxygen levels (SpO2), while the LM35 sensor monitors body temperature. The ESP32 processes this data and displays it locally on an LCD screen for immediate reference. Simultaneously, the data is transmitted over Wi-Fi to the ThingzMate Cloud, where it can be accessed remotely by healthcare providers or caregivers.

The system is designed to provide alerts when any of the monitored parameters exceed safe limits, ensuring that medical attention can be provided promptly. Additionally, it includes security features such as an authorization mechanism to protect sensitive health information from unauthorized access.

By integrating IoT technology with health monitoring systems, this project offers a versatile and scalable solution that enhances patient care. It is particularly useful in scenarios like home healthcare, remote patient monitoring, or emergency medical situations, where real-time data and remote access can make a significant difference in patient outcomes.

Hardware and Software Requirements

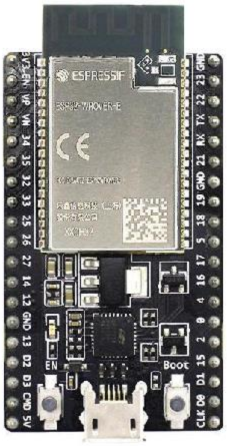
Hardware Requirements

1. ESP32 Microcontroller
2. MAX30105
3. LM35
4. LED
5. BreadBoard
6. USB Cable
7. Jumper Wires

Software Requirements

1. Wokwi Simulator
2. Arduino IDE
3. Thingzmate Cloud

ESP32 Microcontroller

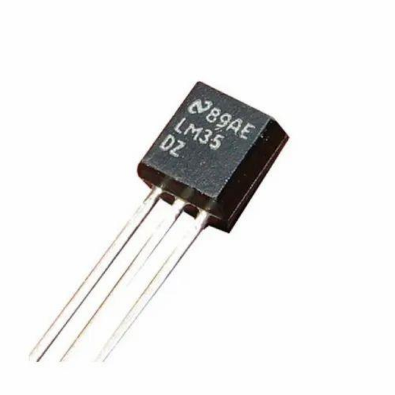


The ESP32 microcontroller is a powerful and versatile chip used in this project to control the four-way traffic light system. It features integrated Wi-Fi and Bluetooth capabilities, allowing for seamless communication with the ThingzMate Cloud for remote monitoring and control. With its dual-core processor and multiple GPIO pins, the ESP32 efficiently handles the timing and sequencing of the traffic lights, making it ideal for real-time IoT applications.

MAX30105

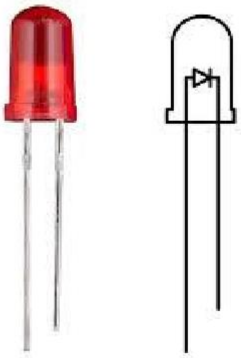
The MAX30105 is a versatile sensor that is primarily used for measuring heart rate, blood oxygen levels (SpO2), and detecting presence. In this health monitoring system, the MAX30105 plays a crucial role in providing accurate and real-time data on the patient's cardiovascular health. It operates by emitting light through an LED and measuring the reflected light from blood vessels, allowing it to calculate heart rate and SpO2. The ESP32 microcontroller processes this data and sends it to the ThingzMate Cloud for remote monitoring, enabling healthcare providers to keep track of vital signs continuously. The sensor's reliability and precision make it a key component in the system, ensuring that the health metrics are monitored effectively.

LM35



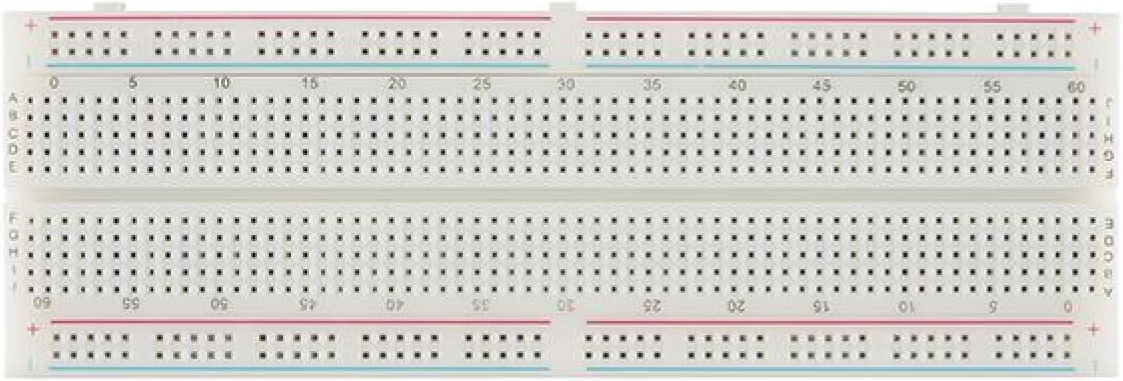
The LM35 is a precision temperature sensor used in this health monitoring system to measure the patient's body temperature. Known for its accuracy, the LM35 provides a linear voltage output that is directly proportional to the temperature in degrees Celsius. The ESP32 reads this output and, after processing, displays the temperature data locally on an LCD screen while also transmitting it to the ThingzMate Cloud. This allows for real-time monitoring and alerts if the body temperature exceeds normal ranges, which is crucial for detecting fever or other health issues early. The LM35's integration into the system ensures that temperature data is captured accurately and consistently, enhancing the overall reliability of the health monitoring system.

LED



I In this IoT-based health monitoring system, standard LEDs are used as indicators for the status of the monitored health parameters. These LEDs, controlled by the ESP32 microcontroller, light up to signal different conditions, such as a normal range or an alert when a parameter exceeds a critical threshold. For example, a green LED might indicate that all vital signs are within safe limits, while a red LED could signify that immediate attention is required. The LEDs provide a simple and effective visual cue, ensuring that health status can be quickly assessed at a glance, both locally and remotely.

BreadBoard



In this IoT-based health monitoring system, the breadboard is a crucial component used for prototyping the circuit connections. It provides a convenient platform for assembling and testing the connections between the ESP32 microcontroller, sensors like the MAX30105 and LM35, and other components such as LEDs and resistors. The breadboard’s flexibility allows for easy modifications to the circuit without soldering, enabling quick adjustments as the design evolves. This makes it an essential tool for iterating and refining the system before finalizing the design.

USB-Cable

The USB cable is a critical tool in this project, used to connect the ESP32 microcontroller to a computer for power supply, programming, and debugging. It enables the transfer of code and data between the development environment and the microcontroller, facilitating the upload of firmware and real-time communication during the development process. The USB connection also allows for serial monitoring, providing valuable insights into the system's performance and behavior.

Jumper Wires

Jumper wires are essential in this project, used to connect the ESP32 microcontroller to the components on the breadboard. These wires provide a flexible and reliable way to link the microcontroller’s GPIO pins to the LEDs, resistors, and other circuit elements, enabling proper signal and power flow. Their ease of use allows for quick modifications and testing during the prototyping stage.

Wokwi Simulator

The Wokwi Simulator is an invaluable online tool used in this IoT-based health monitoring system project to simulate the entire setup before physical deployment. It allows for virtual testing of the ESP32 microcontroller, sensors such as the MAX30105 and LM35, and other components like LEDs. With Wokwi, developers can visualize real-time data output, test different scenarios, and debug the system efficiently. This ensures the functionality of the health monitoring system and enables necessary adjustments to the circuit and code without requiring the actual hardware setup, saving both time and resources during development.

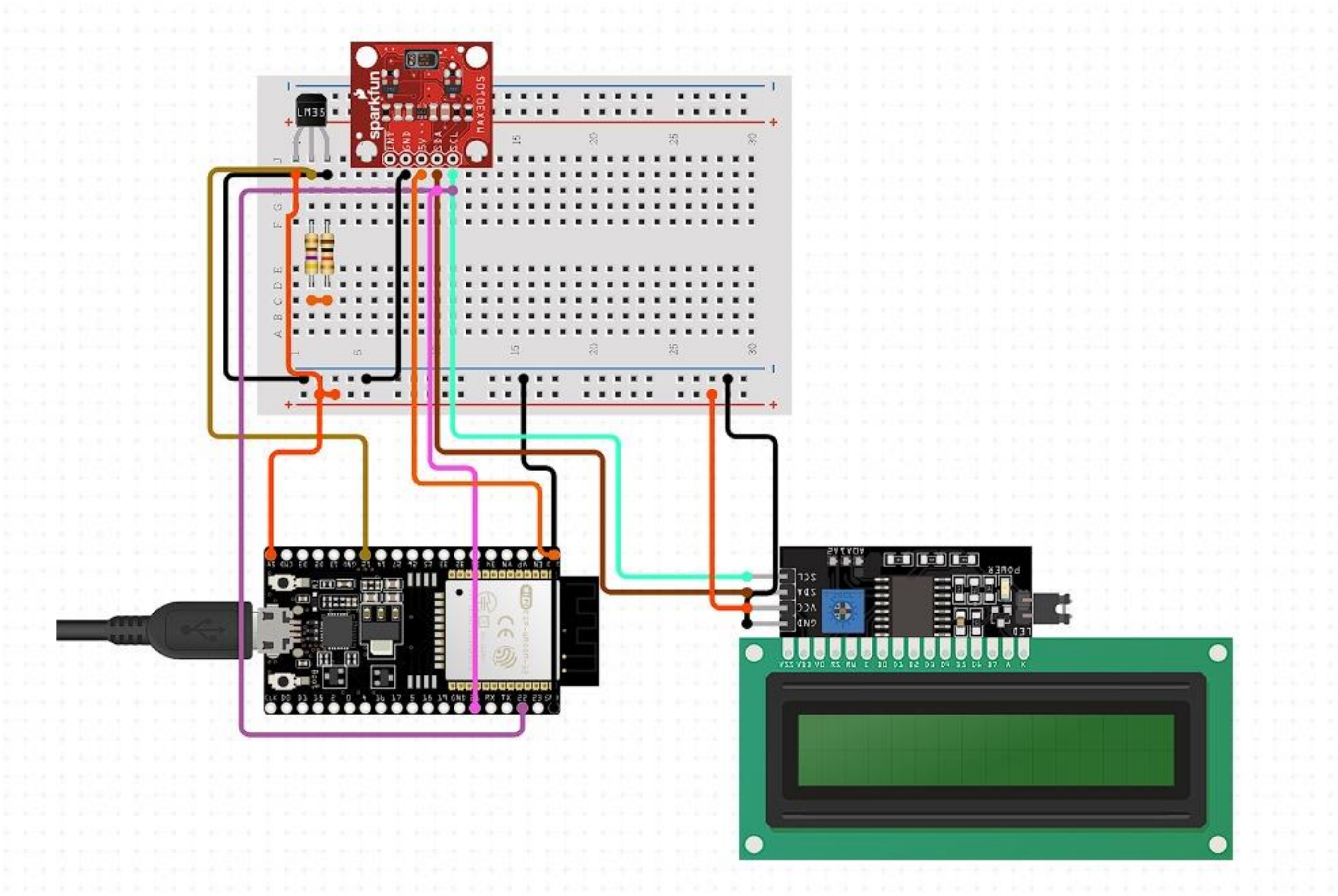
Arduino IDE

The Arduino IDE is the primary development environment used in this IoT-based health monitoring system project for programming and uploading code to the ESP32 microcontroller. It offers a user-friendly interface for writing, compiling, and debugging the code that manages the sensors and data transmission. With extensive library support and seamless compatibility with the ESP32, the Arduino IDE simplifies the development process, enabling efficient code iteration, testing, and deployment.

Thingzmate Cloud

ThingzMate Cloud plays a crucial role in the IoT-based health monitoring system by enabling real-time monitoring and management of health data via cloud connectivity. It provides tools for configuring alerts, visualizing health metrics, and securely storing patient data. With ThingzMate, healthcare providers can remotely access vital signs, ensuring timely interventions and continuous care. The cloud platform also supports data analysis, allowing for insights into health trends, making it an essential component for optimizing patient monitoring and care scenarios.

Block Diagram



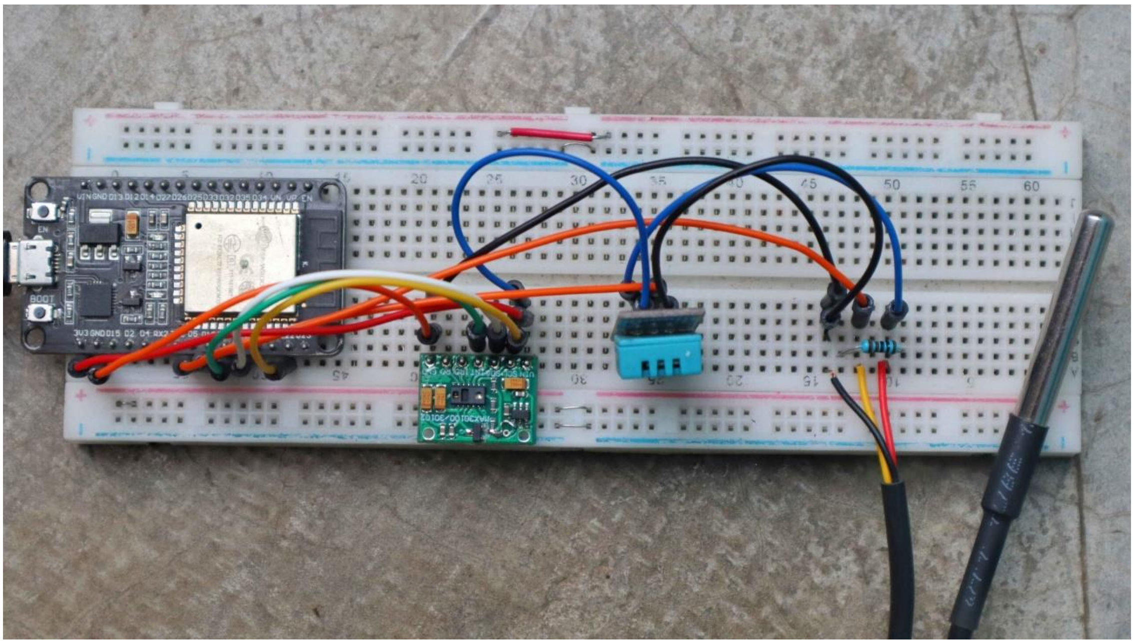
}

Code

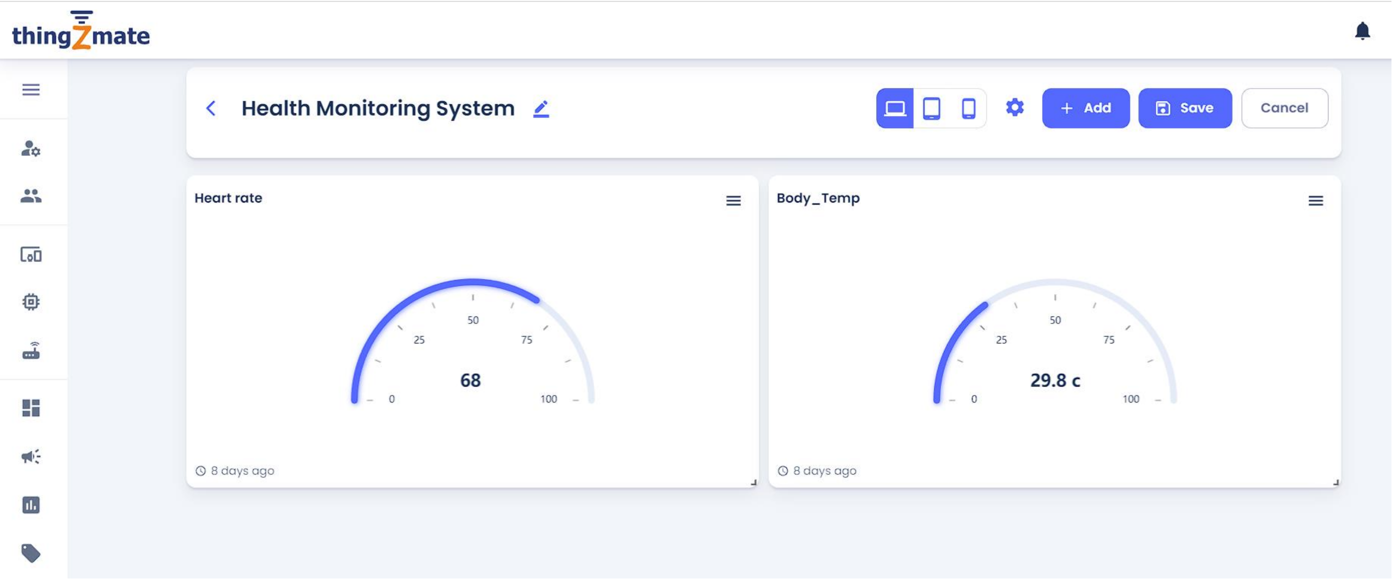
```
#include <Wire.h>
#include "Adafruit_MAX30100.h"
#include <WiFi.h>
#include <HTTPClient.h>

#define LM35_PIN 34

const char* ssid = "Realme";
const char* password = "17022020";

const char *serverUrl = "https://console.thingzmate.com/api/v1/device-types/satheesh101/devices/dev100/uplink"; // Replace with your server endpoint
String AuthorizationToken = "Bearer 7350cfee91bab30d2405b8715826dcdd";
Adafruit_MAX30100 max30100;

float temperatureC;
float heartRate;
float spo2;

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }

  Serial.println("Connected to WiFi");
  max30100.begin();
  max30100.setMode(MAX30100_MODE_SPO2_HR);
  max30100.setHighresModeEnabled(true);

  pinMode(LM35_PIN, INPUT);
}

void loop() {
  readSensors();
  sendToThingzmate();
  delay(10000);
}

void readSensors() {
  int sensorValue = analogRead(LM35_PIN);
  temperatureC = (sensorValue * 3.3 / 4095.0) * 100.0;

  max30100.update();
  heartRate = max30100.getHeartRate();
  spo2 = max30100.getSpO2();

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

  Serial.print("Heart Rate: ");
  Serial.print(heartRate);
  Serial.println(" BPM");
  Serial.print("SpO2: ");
  Serial.print(spo2);
  Serial.println(" %");
}
```

```
void sendToThingzmate() {
  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;
    String url = serverUrl + "devices/" + deviceId + "/datapoints";

    String jsonData = "{";
    jsonData += "\"temperature\":" + String(temperatureC) + ",";
    jsonData += "\"heart_rate\":" + String(heartRate) + ",";
    jsonData += "\"spo2\":" + String(spo2);
    jsonData += "}";

    http.begin(url);
    http.addHeader("Content-Type", "application/json");
    http.addHeader("api-key", apiKey);

    int httpResponseCode = http.POST(jsonData);

    if (httpResponseCode > 0) {
      String response = http.getString();
      Serial.println(httpResponseCode);
      Serial.println(response);
    } else {
      Serial.print("Error code: ");
      Serial.println(httpResponseCode);
    }

    http.end();
  } else {
    Serial.println("WiFi Disconnected");
  }
}
```

Output Results



Cloud Output



Conclusion

The IoT-based health monitoring system developed in this project demonstrates the potential of integrating advanced sensor technology with cloud-based platforms like ThingzMate to deliver real-time, remote healthcare solutions. By leveraging the ESP32 microcontroller, MAX30105 sensor, LM35 sensor, and other components, the system provides continuous monitoring of vital health parameters such as heart rate, blood oxygen levels, and body temperature. The data is securely transmitted to the cloud, allowing healthcare providers and caregivers to monitor patients remotely and respond quickly to any critical changes.

The system's ability to trigger alerts when parameters exceed safe limits ensures timely interventions, which is crucial for improving patient outcomes. Additionally, the use of the Wokwi Simulator and Arduino IDE during development enabled efficient testing and iteration, contributing to a robust and reliable design.