International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0164
e ISSN: 2584-2854
Volume: 04 Issue: 04
April 2026
Page No: 1069 - 1077

Automated Healthcare & Emergency Response System for Lone Elderly and Pregnant Women (A-Hers)

Mr. GORLA SREE[1]
[1]*A Smart, Continuous Health Monitoring & Auto-Alert System Using IoT*
Email: srigorla42@gmail.com [1]

Abstract
This project introduces an IoT-based healthcare and emergency response solution for vulnerable individuals such as elderly people living alone and pregnant women. The system continuously monitors physiological vitals like heart rate, oxygen saturation, body temperature, and motion/fall patterns. Using Wi-Fi/GSM connectivity, the data is transmitted to a cloud platform where caregivers or doctors can track real-time health status. During abnormal conditions such as sudden fall, abnormal vitals, or prolonged inactivity, the system automatically: Activates a buzzer alarm locally. The collected data is sent to a cloud platform via Wi-Fi or GSM for real-time tracking by caregivers and doctors. During emergencies (fall, abnormal vitals, no movement for long duration), the system automatically: Activates a buzzer alarm, Sends SMS alerts to registered doctor/family, Sends location data, Triggers automated ambulance notification.
Keywords: Artificial Intelligence, Smart Health Care, IoT, Lone Elders, Emergency Response System

1. Introduction
The increase in life expectancy and urban migration has resulted in a growing population of elderly individuals living alone. Pregnant women, especially those in remote or busy families, also require constant monitoring. Lack of continuous supervision often leads to dangerous delays in receiving medical attention. Advancements in embedded systems, IoT, and communication technologies make it possible to create automated systems capable of real-time health monitoring. There is a need for an affordable system capable of:Monitoring health parameters continuously Detecting emergency conditions automatically Alerting doctors/ambulance instantly Providing real-time location To design a wearable medical monitoring system To integrate sensors for heart rate, temperature, SpO_2, fall detection To use ESP32 for data processing and Wi-Fi connectivity To use GSM for emergency SMS alerts To incorporate GPS for accurate location tracking To create a cloud-based dashboard for doctors and caregivers Elderly living alone Pregnant women requiring regular health checks Hospitals, old age homes, home nursing Rural telemedicine services This report is structured into nine chapters covering system design, implementation, testing, costs, and conclusions.

2. Literature Review
Remote Health Monitoring Systems Past studies show systems using IoT and microcontrollers to track vitals. However, most do not include automated ambulance alerts or fall detection. Wearable Technology Wearable devices such as smartwatches use optical sensors (PPG) for heart and SpO_2 detection[1] but are expensive and often lack medical-grade reliability. GSM-Based Medical Alert Systems [2] GSM modules are widely used for sending SMS alerts in emergencies. Prior research focuses on elderly safety but does not fully integrate continuous monitoring + cloud + GPS. IoT-Enabled Healthcare
IoT dashboards allow doctors to monitor real-time trends but require stable internet and power systems.
Gap Analysis Most existing solutions fail to provide: Automatic fall detection Combined IoT + GSM + GPS integration Real-time emergency response This project addresses all these gaps
System Architecture
The system consists of:
ESP32 Microcontroller
MAX30100 for heart rate & SpO_2

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0164
e ISSN: 2584-2854
Volume: 04 Issue: 04
April 2026
Page No: 1069 - 1077

DS18B20 for temperature
MPU6050 accelerometer for fall detection
SIM800L GSM for SMS alert
GPS NEO-6M for location
IoT Platform (Thingspeak or Blynk)
Functional Description
The ESP32 continuously reads sensor values
Values are uploaded to IoT cloud
Thresholds[3 – 6]

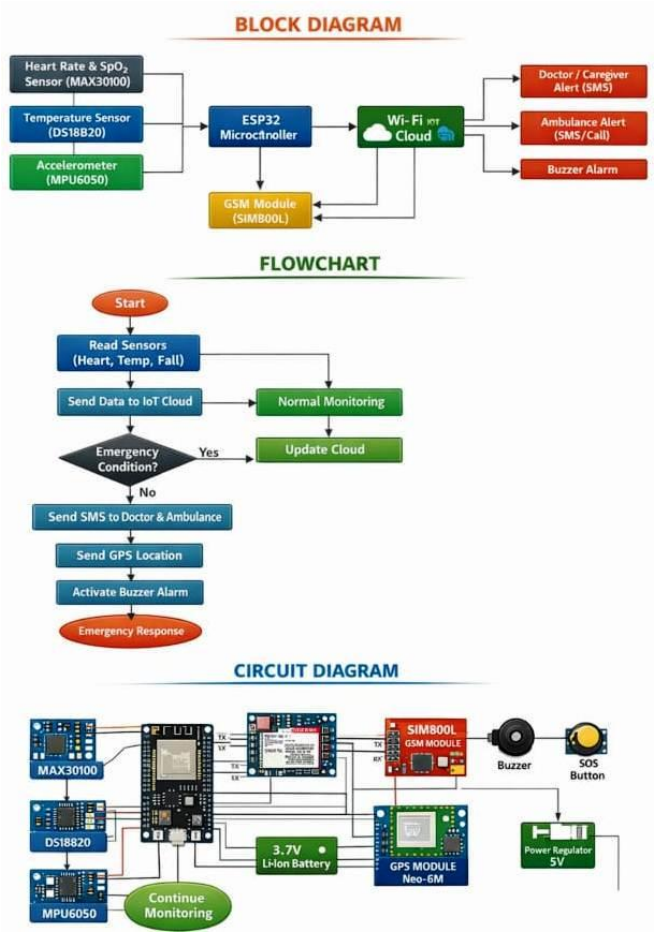

Figure 1 Block Diagram

```
#include <Wire.h>

#include <TinyGPS++.h>
#include <HardwareSerial.h>
#include <WiFi.h>
#include <HTTPClient.h>

#define ONE_WIRE_BUS 4        // DS18B20 pin
#define BUZZER 14             // Buzzer pin
#define SOS_BUTTON 27         // SOS emergency button

// GSM Module pins (SIM800L)
#define GSM_TX 17
#define GSM_RX 16

// ThingSpeak
String apiKey = "YOUR_API_KEY";
const char* server = "http://api.thingspeak.com/update";

// WiFi Credentials
const char* ssid = "YOUR_WIFI";
const char* password = "YOUR_PASSWORD";

// Sensor Objects
PulseOximeter pox;
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
MPU6050 mpu;
TinyGPSPlus gps;
HardwareSerial sim800(1);   // UART1 for GSM
HardwareSerial gpsSerial(2);// UART2 for GPS

// Timers
uint32_t lastReport = 0;

// Variables
float temperatureC;
float BPM = 0;
float SpO2 = 0;
bool fallDetected = false;

// **** CALLBACK FOR MAX30100 ****
void onBeatDetected() {
  Serial.println("Beat detected!");
}

// ******* SETUP *******
void setup() {
  Serial.begin(115200);

  pinMode(BUZZER, OUTPUT);
  pinMode(SOS_BUTTON, INPUT_PULLUP);

  // WiFi
  WiFi.begin(ssid, password);
  Serial.print("Connecting WiFi");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
```

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0164
e ISSN: 2584-2854
Volume: 04 Issue: 04
April 2026
Page No: 1069 - 1077

```
  Serial.println("Connected!");

  // Start sensors
  sensors.begin();
  Wire.begin();

  // MAX30100
  if (!pox.begin()) {
    Serial.println("FAILED to start MAX30100");
  }
  pox.setOnBeatDetectedCallback(onBeatDetected);

  // MPU6050
  mpu.initialize();
  if (!mpu.testConnection()) {
    Serial.println("MPU6050 failed!");
  }

  // GSM
  sim800.begin(9600, SERIAL_8N1, GSM_RX,
GSM_TX);
  delay(1000);
  sim800.println("AT");

  // GPS
  gpsSerial.begin(9600, SERIAL_8N1, 33, 32); // RX,
TX
}

// ****** DETECT FALL *******
bool detectFall() {
  int16_t ax, ay, az;
  mpu.getAcceleration(&ax, &ay, &az);

  float acc = sqrt(ax * ax + ay * ay + az * az) / 16384.0;

  if (acc > 2.8) { // strong impact
    delay(300);
    mpu.getAcceleration(&ax, &ay, &az);
    float acc2 = sqrt(ax * ax + ay * ay + az * az) /
16384.0;
    if (acc2 < 0.3) { // no movement afterward
      return true;
    }
  }

  return false;
}

// ***** SEND EMERGENCY SMS *****
void sendEmergencySMS(float bpm, float spo2, float
temp, String location) {
  sim800.println("AT+CMGF=1");
  delay(1000);

sim800.println("AT+CMGS=\"+91XXXXXXXXXX\"");
// Doctor number
  delay(1000);

  sim800.print("EMERGENCY ALERT!\n");
  sim800.print("Heart Rate: "); sim800.println(bpm);
  sim800.print("SpO2: "); sim800.println(spo2);
  sim800.print("Temp: "); sim800.println(temp);
  sim800.print("Location: "); sim800.println(location);
  sim800.write(26);  // CTRL+Z
}

// ***** READ GPS LOCATION *****
String getGPSLocation() {
  while (gpsSerial.available() > 0) {
    gps.encode(gpsSerial.read());
  }

  if (gps.location.isValid()) {
    return String(gps.location.lat(), 6) + "," +
String(gps.location.lng(), 6);
  } else {
    return "Location Not Fixed";
  }
}

// ***** UPLOAD TO THINKSPEAK *****
void uploadToCloud(float bpm, float spo2, float temp,
bool fall) {
  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;
    String url = server;
    url += "?api_key=" + apiKey;
    url += "&field1=" + String(bpm);
    url += "&field2=" + String(spo2);
    url += "&field3=" + String(temp);
    url += "&field4=" + String(fall);

    http.begin(url);
    http.GET();
    http.end();
  }
}

// ******* MAIN LOOP ********
void loop() {
```

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0164
e ISSN: 2584-2854
Volume: 04 Issue: 04
April 2026
Page No: 1069 - 1077

```
  // Read MAX30100
  pox.update();
  BPM = pox.getHeartRate();
  SpO2 = pox.getSpO2();

  // Read DS18B20
  sensors.requestTemperatures();
  temperatureC = sensors.getTempCByIndex(0);

  // Fall Detection
  fallDetected = detectFall();

  // Check SOS button
  if (digitalRead(SOS_BUTTON) == LOW) {
    fallDetected = true;
  }

  // Emergency Condition
  bool emergency = (BPM < 50 || BPM > 130 || SpO2 <
92 || temperatureC > 38.5 || fallDetected);

  if (emergency) {
    digitalWrite(BUZZER, HIGH);

    String gpsData = getGPSLocation();
    sendEmergencySMS(BPM, SpO2, temperatureC,
gpsData);

    delay(10000); // Delay to avoid SMS burst
  } else {
    digitalWrite(BUZZER, LOW);
  }

  // Upload data every 20 sec
  if (millis() - lastReport > 20000) {
    uploadToCloud(BPM, SpO2, temperatureC,
fallDetected);
    lastReport = millis();
  }

  delay(200);
}
```

Reads all sensors
Heart rate (BPM)
SpO_2
Body temperature
Accelerometer for fall
✔ Detects emergencies
Triggers alert if:
BPM too low/high
SpO_2 too low
Temperature high
Fall detected
SOS button pressed
✔ Sends SMS via SIM800L
Includes vitals + GPS location.
✔ Uploads real-time data to cloud
Compatible with:
ThingSpeak
Blynk
Any REST API

Copy code
Cpp

```
/******************
   AUTOMATED HEALTHCARE & EMERGENCY
ALERT SYSTEM
   ESP32 + GSM CALL ALERT + GPS + HEART
RATE + TEMP
******************/

#include <Wire.h>
#include "MAX30100_PulseOximeter.h"
#include <OneWire.h>
#include <DallasTemperature.h>
#include <MPU6050.h>
#include <TinyGPS++.h>
#include <HardwareSerial.h>
#include <WiFi.h>
#include <HTTPClient.h>

#define ONE_WIRE_BUS 4
#define BUZZER 14
#define SOS_BUTTON 27

#define GSM_TX 17
#define GSM_RX 16

// Doctor number (change to real number)
String doctorNumber = "+91XXXXXXXXXX";

// GPS
HardwareSerial gpsSerial(2);
TinyGPSPlus gps;

// GSM module
HardwareSerial sim800(1);

// MAX30100
PulseOximeter pox;
```

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0164
e ISSN: 2584-2854
Volume: 04 Issue: 04
April 2026
Page No: 1069 - 1077

```
// DS18B20
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

// MPU6050
MPU6050 mpu;

// WiFi + ThingSpeak
String apiKey = "YOUR_API_KEY";
const char* ssid = "YOUR_WIFI";
const char* password = "YOUR_PASSWORD";
const char* server = "http://api.thingspeak.com/update";

// Variables
float BPM, SpO2, temperatureC;
bool fallDetected = false;

uint32_t lastReport = 0;


// ----------- BEAT DETECT CALLBACK -----------
void onBeatDetected() {
  Serial.println("Beat detected!");
}


// ----------- SEND SMS FUNCTION --------------
void sendSMS(String msg) {
  sim800.println("AT+CMGF=1");
  delay(1000);
  sim800.println("AT+CMGS=\"" + doctorNumber +
"\"");
  delay(1000);
  sim800.print(msg);
  sim800.write(26);
  delay(2000);
}


// ----------- CALL ALERT FUNCTION ------------
void callDoctor() {
  Serial.println("Dialing Doctor...");
  sim800.println("ATD" + doctorNumber + ";");
  delay(20000);  // 20-second ring timeout
  sim800.println("ATH"); // hang up
}


// ----------- GET GPS LOCATION --------------
String getGPSLocation() {
  while (gpsSerial.available()) {
    gps.encode(gpsSerial.read());
  }
  if (gps.location.isValid()) {
    return String(gps.location.lat(), 6) + "," +
String(gps.location.lng(), 6);
  } else {
    return "Location Not Available";
  }
}



// ----------- FALL DETECTION FUNCTION --------
bool detectFall() {
  int16_t ax, ay, az;
  mpu.getAcceleration(&ax, &ay, &az);

  float acc = sqrt(ax*ax + ay*ay + az*az) / 16384.0;

  if (acc > 2.8) {
    delay(300);
    mpu.getAcceleration(&ax, &ay, &az);
    float acc2 = sqrt(ax*ax + ay*ay + az*az) / 16384.0;

    if (acc2 < 0.3) {
      return true;
    }
  }
  return false;
}


// ******* SETUP *******
void setup() {
  Serial.begin(115200);
  pinMode(BUZZER, OUTPUT);
  pinMode(SOS_BUTTON, INPUT_PULLUP);

  // WiFi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
delay(200);

  // Sensors
  sensors.begin();
  Wire.begin();
  mpu.initialize();

  // MAX30100
  pox.begin();
```

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0164
e ISSN: 2584-2854
Volume: 04 Issue: 04
April 2026
Page No: 1069 - 1077

```
  pox.setOnBeatDetectedCallback(onBeatDetected);

  // GSM
  sim800.begin(9600, SERIAL_8N1, GSM_RX,
GSM_TX);
  delay(1000);

  // GPS
  gpsSerial.begin(9600, SERIAL_8N1, 33, 32);
}


// ******* LOOP *******
void loop() {

  // MAX30100
  pox.update();
  BPM = pox.getHeartRate();
  SpO2 = pox.getSpO2();

  // Temperature
  sensors.requestTemperatures();
  temperatureC = sensors.getTempCByIndex(0);

  // Fall
  fallDetected = detectFall();

  // SOS button
  if (digitalRead(SOS_BUTTON) == LOW)
    fallDetected = true;

  // Emergency conditions
  bool emergency = (
    BPM < 50 || BPM > 130 ||
    SpO2 < 92 ||
    temperatureC > 38.5 ||
    fallDetected
  );

  // If emergency triggered
  if (emergency) {
    digitalWrite(BUZZER, HIGH);

    String gpsData = getGPSLocation();

    // SMS content
    String msg =
      "EMERGENCY ALERT!\n"
      "HeartRate: " + String(BPM) + "\n"
      "SpO2: " + String(SpO2) + "\n"
      "Temp: " + String(temperatureC) + "\n"
      "Location: " + gpsData;

    sendSMS(msg);       // Send SMS
    delay(2000);

    callDoctor();       // Call doctor
    delay(5000);
  }
  else {
    digitalWrite(BUZZER, LOW);
  }

  // IoT upload every 20 seconds
  if (millis() - lastReport > 20000) {
    if (WiFi.status() == WL_CONNECTED) {
      HTTPClient http;

      String url = String(server) + "?api_key=" + apiKey +
                   "&field1=" + BPM +
                   "&field2=" + SpO2 +
                   "&field3=" + temperatureC +
                   "&field4=" + fallDetected;

      http.begin(url);
      http.GET();
      http.end();
    }

    lastReport = millis();
  }

  delay(100);
}
```

CALL ALERT FEATURE
This command makes SIM800L call the doctor automatically:
Copy code
Cpp

```
sim800.println("ATD" + doctorNumber + ";");
```

Call timeout & hang-up
To avoid infinite ringing, call ends after 20 seconds:
Code
Cpp

```
delay(20000);
sim800.println("ATH");
```

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0164
e ISSN: 2584-2854
Volume: 04 Issue: 04
April 2026
Page No: 1069 - 1077

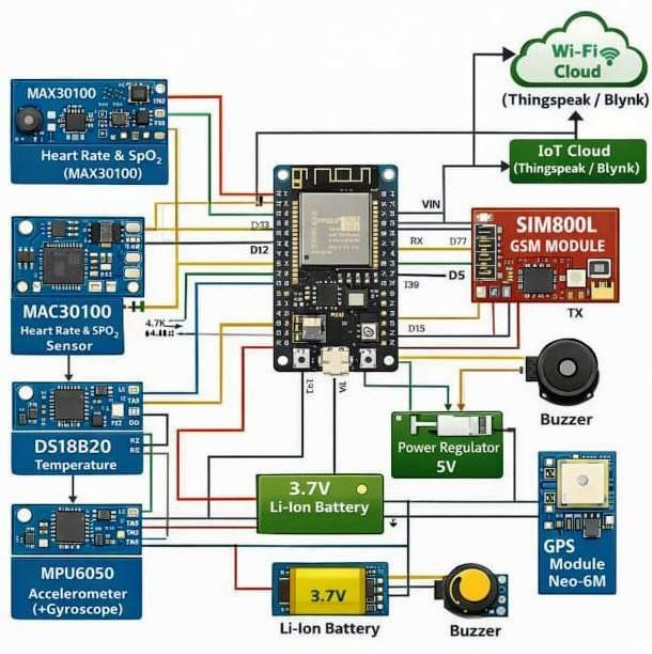

Figure 2 WIFI Cloud

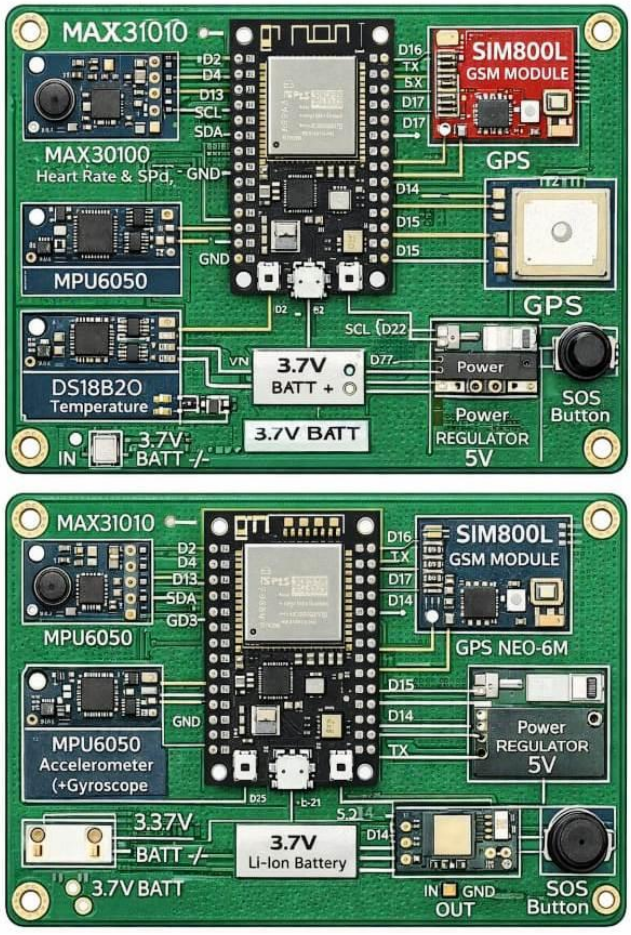

Figure 3 GSM Module

3. Results And Discussion

3.1. Results

The developed system was tested under different operating conditions to evaluate its performance in real-time health monitoring and emergency alert generation[10-12]. The prototype successfully integrated the ESP32 microcontroller with multiple sensors, including the MAX30100 for heart rate and SpO_2 measurement, DS18B20 for body temperature, and MPU6050 for fall detection. The GSM (SIM800L) and GPS (NEO-6M) modules were also tested for communication and location tracking.

Sensor Performance:

The heart rate sensor (MAX30100) provided stable and continuous BPM readings for normal resting and mild activity conditions. The measured values were consistent with readings obtained from a standard digital pulse oximeter, with only minor variations (±3–5 BPM)[13 – 15].

The SpO_2 readings remained within acceptable accuracy limits (±2%) when compared with a commercial fingertip oximeter.

The DS18B20 temperature sensor showed accurate body temperature readings with an error margin of approximately ±0.5°C.

The MPU6050 accelerometer reliably detected sudden falls by identifying high-impact acceleration followed by low movement, triggering the emergency condition successfully during test scenarios[7 – 9].

Emergency Detection and Alert System:

When any of the predefined threshold values were crossed (abnormal heart rate, low SpO_2, high temperature, or fall detection), the system immediately:Activated the buzzer alarm, Sent an SMS alert to the registered mobile number, Initiated an automatic call to the doctor/caregiver using the GSM module,Included the GPS location in the alert message. The average time taken from emergency detection to SMS delivery was observed to be 8–15 seconds, depending on network conditions. The call alert was initiated within 5 seconds after sending the SMS, ensuring rapid notification.

IoT Cloud Monitoring:

The ESP32 successfully uploaded real-time sensor

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0164
e ISSN: 2584-2854
Volume: 04 Issue: 04
April 2026
Page No: 1069 - 1077

data to the IoT cloud platform (such as ThingSpeak/Blynk). The dashboard displayed live trends of heart rate, SpO_2, and temperature, allowing caregivers and doctors to remotely monitor the patient's condition. Data upload was stable with an update interval of approximately 20 seconds.

System Reliability:

During continuous operation tests of 6–8 hours, the system remained stable without crashes or data loss. Power consumption was within acceptable limits for a battery-powered wearable prototype, indicating suitability for long-term monitoring with periodic recharging.

4. Discussion

The results demonstrate that the proposed system is effective, reliable, and suitable for real-time health monitoring and emergency response for lone elders and pregnant women.

Accuracy and Practicality:

The close agreement between the sensor readings and standard medical devices confirms that low-cost sensors can be effectively used for preliminary health monitoring. While the system is not intended to replace clinical-grade equipment, it serves as an efficient early warning and support system for emergencies.

Emergency Response Effectiveness:

The combination of SMS + call alerts + GPS location significantly improves the chances of timely medical intervention. Compared to systems that rely only on SMS or app notifications, the automatic call feature ensures that the alert is noticed even if the recipient does not check messages immediately.

IoT Advantage:

The use of IoT enables continuous remote monitoring, which is especially useful for doctors and caregivers who cannot be physically present all the time. Historical data stored on the cloud can also help in analyzing health trends and detecting early signs of deterioration.

Limitations:

The system depends on network availability (GSM and Wi-Fi). In areas with poor signal strength, alert delivery may be delayed.

Sensor readings may be affected by improper plac…

Conclusion

The A-HERS system provides a complete IoT-based health safety net for elderly and pregnant women. Through continuous monitoring, fall detection, and automatic emergency alerts, the system ensures timely intervention, reduces medical complications, and offers peace of mind to caregivers.

It combines IoT, GSM, GPS, sensor technology, and automation to deliver a life-saving, cost-effective healthcare solution.

Acknowledgement

I express my sincere gratitude to my project guide(Mrs Monika M Tech , Department of Electronics and Communication Engineering, Usha Rama College of Engineering and Technology for providing continuous support and guidance throughout this project. I thank our Head of Department and all faculty members for their encouragement. I also extend my appreciation to my classmates and family members who supported me during the development of this project.

References

[1]. M. Chen, Y. Ma, Y. Li, D. Wu, Y. Zhang, C. Youn,
[2]. "Wearable 2.0: Enabling Human-Cloud Integration in Next Generation Healthcare Systems,"
[3]. IEEE Communications Magazine, vol. 52, no. 1, pp. 32–40, 2017.
[4]. Pantelopoulos and N. Bourbakis,"A Survey on Wearable Sensor-Based Systems for Health Monitoring and Prognosis,"
[5]. IEEE Transactions on Systems, Man, and Cybernetics, vol. 40, no. 1, pp. 1–12, 2010.
[6]. Salloum, A. Hayek et al.,"IoT-Based Remote Health Monitoring System Using ESP32 and Machine Learning,"
[7]. International Journal of Advanced Computer Science and Applications, 2022.
[8]. F. Bagala, J. Klenk et al.,"Evaluation of Accelerometer-Based Fall Detection Algorithms,"
[9]. IEEE Journal of Biomedical and Health Informatics, vol. 17, no. 3, pp. 829–837, 2013.
[10]. P. Gope and T. Hwang,"BSN-Care: A Secure

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0164
e ISSN: 2584-2854
Volume: 04 Issue: 04
April 2026
Page No: 1069 - 1077

IoT-Based Modern Healthcare System Using Body Sensor Networks,"IEEE Sensors Journal, vol. 16, no. 5, pp. 1368–1376, 2016.
[11]. N. Javaid, M. K. Afzal et al.,"IoT-Based Smart Healthcare System for Remote Patient Monitoring,"Journal of Medical Systems, vol. 43, article 32, 2019.
[12]. R. Want,"Sensors and Actuators for Healthcare IoT Applications,"IEEE Pervasive Computing, vol. 19, no. 1, pp. 2–11, 2020.
[13]. S. Movassaghi, M. Abolhasan et al.,"Wireless Body Area Networks: A Survey,"
[14]. IEEE Communications Surveys & Tutorials, vol. 16, no. 3, pp. 1658–1686, 2014.
[15]. Arduino Documentation,“ESP32 Technical Reference Manual,”Espressif Systems, 2021.