Abstract

Musculoskeletal disorders and poor posture are becoming increasingly common due to sedentary lifestyles. Traditional physiotherapy requires frequent hospital visits, which are time-consuming and expensive. Furthermore, manual monitoring of muscle fatigue is prone to human error, and patients often neglect their exercises or posture correction at home.

This project presents an **IoT Enabled Smart Physio System** that integrates muscle fatigue detection, real-time posture monitoring, and automated TENS (Transcutaneous Electrical Nerve Stimulation) therapy. The system utilizes a Flex Sensor to measure muscle strain and an MPU6050 Gyroscope to detect unsafe body angles. An ESP32 microcontroller processes this data and syncs it to a Google Firebase Realtime Database.

When muscle fatigue exceeds a calibrated threshold, the system automatically triggers a custom-built TENS unit to provide pain relief via electrical impulses. Simultaneously, an Android Application provides a live dashboard for doctors and patients, displaying safety status (Safe/Rolling) and muscle condition (Active/Relaxed). The proposed system offers a low-cost, wearable, and automated solution for modern rehabilitation.

Contents

List of Figures

1. Introduction

1.1 Overview

Physiotherapy is essential for recovery from injuries and managing chronic pain. However, the effectiveness of therapy depends heavily on patient compliance and continuous monitoring. Traditional TENS (Transcutaneous Electrical Nerve Stimulation) machines are standalone devices that lack data logging or intelligent automation. This project bridges that gap by creating a smart, connected device.

1.2 Problem Statement

Current rehabilitation methods face several challenges:

1. **Lack of Monitoring:** Doctors cannot track if a patient is performing exercises correctly at home.
2. **Manual Operation:** Standard TENS units require manual adjustment, which can be difficult for elderly patients.
3. **Safety Risks:** Patients with mobility issues are at risk of falling or rolling over without immediate detection.

1.3 Objectives

The primary objectives of this project are:

- To design a wearable device that detects muscle fatigue using Flex sensors.
- To implement a posture monitoring system using the MPU6050 gyroscope.
- To automate TENS therapy triggering based on real-time sensor data.
- To develop an Android Application for remote monitoring via Google Firebase.
- To ensure patient safety through electrical isolation and brownout protection.

2. System Architecture

2.1 Block Diagram

The system consists of three main layers: the Sensing Layer, the Processing Layer, and the Application Layer.

Figure 2.1: System Block Diagram

2.2 Component Description

- **ESP32 Microcontroller:** The central brain that collects sensor data, runs the safety algorithms, and manages WiFi connectivity.
- **MPU6050:** A 6-axis accelerometer/gyroscope used to detect body tilt and rolling motions.
- **Flex Sensor:** A resistive sensor that changes resistance when bent, used to measure muscle contraction.
- **TENS Unit:** A high-voltage generator circuit driven by the ESP32 to provide therapeutic stimulation.
- **Google Firebase:** A NoSQL cloud database that acts as the bridge between the hardware and the Android App.

3. Hardware Implementation

3.1 Circuit Design

The circuit is divided into two isolated sections: the Low-Voltage Control Logic (ESP32) and the High-Voltage Driver (TENS).

3.1.1 TENS High Voltage Generator

The TENS unit utilizes a 9-0-9 Step-Up Transformer. A 555 Timer IC (or PWM signal from ESP32) drives a **TIP122 Darlington Transistor**, which switches the transformer primary coil. This induces high-voltage pulses on the secondary coil, which are delivered to the patient via electrode pads.

3.1.2 Safety Isolation

To protect the sensitive ESP32 microcontroller from high-voltage spikes, **PC817 Optocouplers** are used. This ensures that there is no direct electrical connection between the microcontroller pins and the high-voltage driver circuit.

3.2 Wiring Connections

The table below details the connections between the sensors and the ESP32.

Table 3.1: ESP32 Pin Connections

Component	**ESP32 Pin**	**Function**
MPU6050 SDA	GPIO 21	I2C Data
MPU6050 SCL	GPIO 22	I2C Clock
Flex Sensor	GPIO 34	Analog Input
Buzzer	GPIO 19	Alarm Output
TENS Trigger	GPIO 4	Therapy Control

4. Software Implementation

4.1 Firmware Logic (ESP32)

The firmware is written in C++ using the Arduino IDE. Key features include:

- **Brownout Protection:** The code disables the standard brownout detector to prevent resets during high current draw from the WiFi module or TENS unit.
- **WiFi Power Optimization:** Transmit power is lowered to 8.5dBm to extend battery life.
- **Sensor Fusion:** Data from the Flex sensor and MPU6050 is processed to determine the ”Fatigue” and ”Posture” states.

4.2 Android Application

The Android app is developed in Java using Android Studio. It connects to the Firebase Realtime Database to fetch live values.

- **Status Indicators:** The UI background changes color (Green for Safe, Red for Unsafe) based on the ‘Posture‘ node in Firebase.
- **Data Logging:** It displays raw sensor values for debugging and calibration.

5. Results and Discussion

5.1 Performance Analysis

The prototype was tested under various conditions.

- **Fatigue Detection:** The flex sensor threshold was calibrated to a value of 15. The system successfully triggers the ”Active” state when the sensor is bent beyond this point.
- **Posture Monitoring:** The MPU6050 accurately detects tilt angles greater than 4000 (raw value), triggering the buzzer and updating the app status to ”Rolling”.

5.2 Screenshots

Figure 5.1: Android App Interface: Safe State (Left) and Warning State (Right)

Conclusion

Summary

The IoT Enabled Smart Physio System successfully demonstrates the integration of medical therapy with modern IoT technology. By automating the TENS trigger mechanism and providing remote monitoring capabilities, the system enhances patient safety and therapy effectiveness. The use of brownout protection and opto-isolation ensures the device is robust enough for portable battery operation.

Future Scope

- **Machine Learning:** Implement edge AI to predict muscle spasms before they occur.
- **PCB Design:** Miniaturize the circuit onto a single SMD PCB for a wristwatch form factor.
- **Telemedicine:** Integrate video calling into the Android app for real-time doctor consultations.

Bibliography

[1] Espressif Systems, “ESP32 Series Datasheet”, 2024.

[2] Google Firebase Documentation, “Realtime Database for Android”, 2024.

[3] InvenSense, “MPU-6000 and MPU-6050 Product Specification”.

[4] ResearchGate, “Transcutaneous Electrical Nerve Stimulation (TENS): Mechanisms and Clinical Application”.

A. Source Code

A.1 ESP32 Firmware (Sketch_for_esp.ino)

```
#include <Wire.h>
#include <WiFi.h>
#include <Firebase_ESP_Client.h>
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"

#define WIFI_SSID "HospitalWiFi"
#define WIFI_PASSWORD "12345678"
#define API_KEY "AIzaSyBkuDFl_I3gG9byZ_OgF292f-v-bIcGhPI"
#define DATABASE_URL "flexa-22635-default-rtdb.firebaseio.com"

void setup() {
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0); // Disable Brownout
  setCpuFrequencyMhz(80); // Save Power
  Serial.begin(115200);
  // ... (Full code omitted for brevity in report)
}

void loop() {
  // Main Logic Loop
}
```

A.2 Android Activity (MainActivity.java)

```
package com.example.flexa;

import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

public class MainActivity extends AppCompatActivity {
    DatabaseReference mDatabase;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mDatabase = FirebaseDatabase.getInstance().getReference();
        // Listener logic here...
    }
}
```