Earthquake Early Warning System

Presented by

Sumit Bhagat (16900323166)
Sourik Banerjee (16900323143)
Subhabrata Mondal (16900323152)
3rd Sem ECE

Problem Statement

- One of the central challenges contributing to the problem was low quality of monitoring and presence of pseudo values.
- Also the challenge stands in the affordability of sensing devices and a centralized network of the sensors located in different regions.
- The critical problem definition revolves around the need for a holistic and technologically empowered approach to enhance the efficiency and effectiveness of an early warning system for earthquakes.

Solution Overview

- Collection of data from a network of nodes containing an ESP32 and a MPU6050 digital accelerometer/gyroscope by detection of the p-wave.
- The esp32 will take the input and send the magnitude of the vibrations recorded to the main backend server.
- A dedicated progressive web application will cater the people and spread the information on the quake in case of the condition the safe threshold limit fails notified by backend and send a notification alert to all the connected users along with the location of the node location.
- Storing the data history for at least for 1 month to analyze the data for later events.

Block Diagram

Data Aquitisition from MPU6050 accelerometer/gyroscope

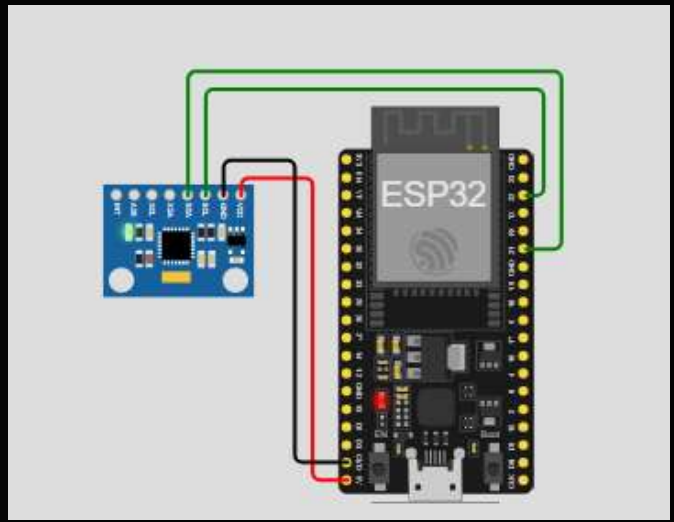
→ Data transfer to ESP32 and calculation of magnitude

→ Passing on the calculated value to backend server

↓ Calculation of risk and analyzing the probability

← Sending notification to all the users of incoming quake with the location and range of the certain ESP32 node

← Saving the history of data for at least 1 month



Circuit Diagram

https://github.com/sumit-bhagat-2004/Pookie_EEWS.git

Hardware Description

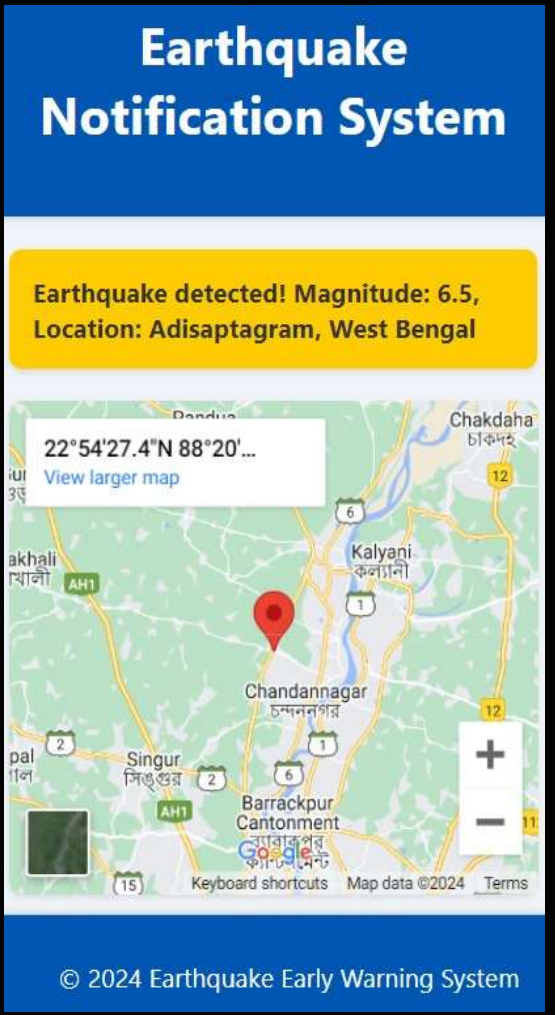
- **MPU6050 accelerometer/gyroscope:**
  - The MPU6050 combines a 3-axis accelerometer and a 3-axis gyroscope on a single chip. In the context of an earthquake warning system, the MPU6050 can be used to detect ground motion by measuring acceleration along X, Y, and Z axis.
  - The MPU6050 uses the I2C communication protocol allows it to interface easily with ESP32.
  - In an earthquake warning system, multiple MPU6050 sensors can be deployed across a region to form a network of monitoring stations.
- **ESP32 Devkit:**
  - The ESP32 is a powerful and versatile microcontroller widely used in IoT applications due to its rich feature set and cost-effectiveness.
  - he ESP32 features integrated Wi-Fi (802.11 b/g/n) with support for both station and access point modes.

Implementation Details

- Simulate and get the details of the X,Y & Z axis from the mpu6050 accelerometer and send it to esp32 through I2C protocol.
- ESP gets the data and performs the formula to get the magnitude of the vibrations or supposedly p- waves.
- It then sends the data to the server through websocket as it connects real time and stays connected.
- The flask backend receives the data over websockets and calculate the impact range and epicenter based on it.
- It stores it in MySQL database for future uses from frontend and sends formatted data to frontend via websockets
- Frontend connects to the websocket and receives the data from backend.

Implementation Details

- It parses the payload to get the epicenter longitude and latitude, magnitude and range.
- It uses service worker to send notification to the users enabled notifications for the progressive web app mentioning magnitude.
- It uses Google map api to locate and show the epicenter on the map.
- The view history button triggers the js function that fetches the data from servers api gateway of SQL table and displays it to the user.



Conclusion

- Used the MPU6050 accelerometer to detect seismic vibrations with mcu ESP32 to process sensor data and communicate with the server using WebSocket protocol for real-time updates.
- Analyze sensor data on the ESP32 to identify high-magnitude P-waves.
- Minimize false alarms by distinguishing between normal vibrations and potential earthquakes.
- Frontend Develop using HTML and CSS for a responsive design that provides real-time updates and alerts and use JavaScript to enhance interactivity and real-time updates on the web app.
- Backend Implement with Flask to handle data processing and server-client communication and use Flask to process incoming data from ESP32 nodes and manage alert notifications.
- Store historical seismic data and user settings for customized alerts.
- Send real-time alerts when significant seismic activity is detected, with options for user customization.
- Deploy multiple ESP32 sensor nodes for wide geographic coverage.

Thank You !