Bachelor of Science in Computer Science and Engineering
CSE 306 (Spring 2025): Project Report

IoT-Based Real-Time Earthquake Detection with Real-Time Telegram Notification and Smart Alert System

Submitted by:
Rasheduzzaman Rakib
CS-D-77-22-120080
Roll: 37

Md. Asraful Molla
CS-D-77-22-120017
Roll: 27

Submitted to:
Md. Nahid Hasan
Lecturer

Department of Computer Science and Engineering
Dhaka International University
Dhaka, Bangladesh

April 2025

Table of Contents

IoT-Based Real-Time Earthquake Detection with Real-Time Telegram Notification and Smart Alert System

1. Abstract

This project presents an IoT-based earthquake detection system designed to provide **instant local alerts and remote notifications** during seismic activity. Utilizing a vibration sensor (SW-420), an ESP8266 microcontroller, a buzzer, and Telegram API, the system enables fast and cost-effective earthquake warning. When vibrations are detected, a buzzer and LED alert nearby individuals, while a real-time message is sent to users via Telegram. This system can serve as a prototype for smart city applications and community-level disaster preparedness solutions.

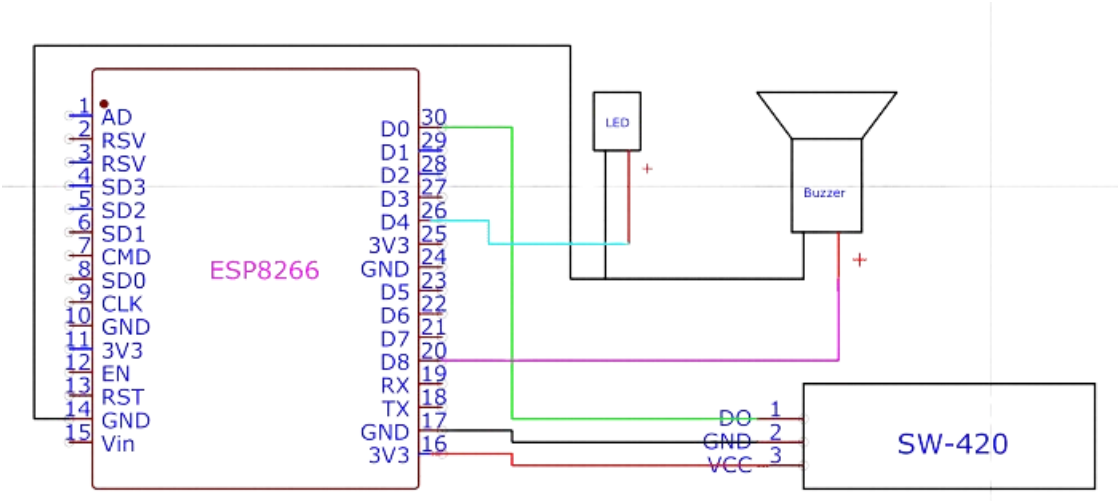
2. Objectives

- To detect ground vibrations using the SW-420 vibration sensor.
- To provide local alerts using a buzzer and LED.
- To send real-time earthquake notifications via Telegram using Wi-Fi (ESP8266).
- To develop a low-cost and efficient early warning system.

3. Components List

Component	Quantity	Description
ESP8266 Wi-Fi Module	1	Microcontroller with built-in Wi-Fi
SW-420 Vibration Sensor	1	Used to detect vibration or shocks
Buzzer	1	Audible alarm indicator
Red LED	1	Visual alert indicator
330-ohm Resistor	1	Current limiting resistor for LED
Breadboard	1	For prototyping circuit without soldering
Jumper Wires	Several	For electrical connections

4. Diagram



5. Working Principle

- The **SW-420 sensor** detects seismic vibrations and outputs a digital signal.
- The **ESP8266 microcontroller** constantly monitors the vibration sensor's output.
- If vibrations are detected, the **buzzer and LED** are activated for visual and audible alerts.
- Simultaneously, a **Telegram message** is sent using the Telegram Bot API via secure HTTPS communication.
- The system waits until the vibration ceases before rearming itself for the next event.

6. Code Summary

The core of the system is programmed in Arduino IDE using the following key functions:

- **WiFi Initialization:**
  Connects ESP8266 to the configured network.
- **Telegram Notification:**
  Sends HTTP GET requests to the Telegram Bot API using WiFi Client Secure.
- **Vibration Detection Loop:**
  Continuously checks for high state from SW-420 and triggers alerts accordingly.

7. Key Features

- Real-time remote notification using **Telegram Bot API**
- Onboard **buzzer and LED alert system** for local signaling
- **Wi-Fi-based**, no SIM card or GSM module required
- Efficient design with **2-second interval throttle** to avoid spamming
- Can be extended with GPS and data logging capabilities

8. Advantages

- **Cost-effective**: Uses affordable and readily available components.
- **Real-time alerts**: Telegram provides instant global message delivery.
- **Scalable**: Can be expanded for large networks of sensors.
- **User-friendly**: Easily programmable and deployable.

9. Limitations and Future Work

Current Limitation	Future Enhancement Proposal
False triggers from environmental vibration	Implement software-based noise filtering or integrate multi-sensor logic (e.g., accelerometer + gyroscope).
No location data	Integrate a GPS module to include real-time latitude and longitude in notifications.
No data storage	Connect to cloud platforms (e.g., Firebase, ThingsBoard, or AWS IoT) for logging and historical analysis.
Manual Telegram setup	Develop a user-friendly mobile/web dashboard to manage bot tokens and chat IDs.
Limited to basic alerts (buzzer, LED, message)	Expand to integrate with smart home systems for **automated emergency response**.
No auto shutdown in critical systems	**Add relays or IoT smart switches to automatically**

10. Conclusion

The developed system successfully demonstrates a **smart, connected, and low-cost solution** for earthquake detection and alerting. Through IoT integration, it enables **real-time responses** to potentially dangerous seismic activity, enhancing safety and awareness. With further refinement, this prototype can be deployed in schools, homes, or public buildings as part of a broader disaster response network.

11. Arduino Code

```
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>

// WiFi credentials
const char *ssid = "###";
const char *password = "###
```

```
// Telegram bot details
const char *botToken = "###";
const char *chatID = "###";

// Define pins
const int vibrationPin = D0; // SW-420 sensor digital output
const int buzzerPin = D8;    // Buzzer pin
const int ledPin = D4;       // LED pin

WiFiClientSecure client;
unsigned long lastAlertTime = 0;
const int alertInterval = 2000;

void setup()
{
 pinMode(vibrationPin, INPUT);  // Set vibration sensor as input
 pinMode(buzzerPin, OUTPUT);    // Set buzzer as output
 pinMode(ledPin, OUTPUT);       // Set LED as output
 digitalWrite(buzzerPin, LOW);  // Start with buzzer off
 digitalWrite(ledPin, LOW);     // Start with LED off

 Serial.begin(115200);

 // Connect to WiFi
 WiFi.begin(ssid, password);
 Serial.print("Connecting to WiFi...");
 while (WiFi.status() != WL_CONNECTED)
 {
  delay(1000);
  Serial.print(".");
 }
 Serial.println("\nConnected to WiFi!");
}

void sendTelegramMessage(String message)
{
 client.setInsecure(); // Bypass SSL certificate
 if (client.connect("api.telegram.org", 443))
 {
  String url = "/bot" + String(botToken) + "/sendMessage?chat_id=" + String(chatID) +
"&text=" + message;
  client.print(String("GET ") + url + " HTTP/1.1\r\n" +
               "Host: api.telegram.org\r\n" +
```

```
            "User-Agent: ESP8266\r\n" +
            "Accept: */*\r\n" +
            "Connection: close\r\n\r\n");
  delay(100);
  while (client.available())
  {
   String response = client.readString();
   Serial.println(response);
  }
  client.stop();
 }
 else
 {
  Serial.println("Failed to connect to Telegram API");
 }
}

void loop()
{
 int vibrationState = digitalRead(vibrationPin);
 unsigned long currentTime = millis();

 if (vibrationState == HIGH)
 {
  Serial.println("Earthquake Detected!");

  // Turn on buzzer and LED for 2 seconds
  for (int i = 0; i < 2; i++)
  {
   digitalWrite(buzzerPin, HIGH);
   digitalWrite(ledPin, HIGH);
   delay(2000);

   // Turn off buzzer and LED
   digitalWrite(buzzerPin, LOW);
   digitalWrite(ledPin, LOW);
   delay(250);
  }

  // Send Telegram alert with interval
  if (currentTime - lastAlertTime >= alertInterval)
  {
   sendTelegramMessage("ভূমিকম্প! ভূমিকম্প! অতি দ্রুত বাসা ত্যাগ করুন!");
```

```
    lastAlertTime = currentTime;
   }

   // Wait until vibration stops before detecting again
   while (digitalRead(vibrationPin) == HIGH)
   {
    delay(5); // Prevents multiple detections from same vibration event
   }
  }
  delay(5); // Small delay for faster response
}
```

12. References

1. ESP8266 Official Documentation
2. Telegram Bot API Documentation
3. SW-420 Vibration Sensor Datasheet
4. Arduino.cc Official Tutorials and Examples
5. NodeMCU (ESP8266) with Arduino IDE Guide
6. Secure HTTPS Communication with ESP8266
7. Telegram Setup with ESP8266

13. User Manual

Version: 1.0

This section provides end-users with instructions for setting up, operating, and maintaining the earthquake detection system.

13.1 Introduction

This system detects ground vibrations using a vibration sensor and sends emergency alerts through Telegram and a local buzzer/LED system. Designed to be simple, real-time, and affordable, it can be deployed in homes, schools, or community centers.

13.2 Package Contents

Item	Quantity	Description
ESP8266 Wi-Fi Module	1	Microcontroller with Wi-Fi
SW-420 Vibration Sensor	1	Detects vibration or shock
Buzzer	1	Local audible alarm
Red LED	1	Visual indicator
330-ohm Resistor	1	Limits LED current
Breadboard	1	For assembling without soldering
Jumper Wires	Several	For making circuit connections
Micro USB Cable	1	For power and programming

13.3 Safety Precautions

- Do not expose the system to moisture or physical damage.

- Handle wiring with care to avoid short circuits.
- Only use 5V regulated power sources.
- Mount the sensor on a solid, non-vibrating base.

13.4 System Overview

- Monitors seismic vibrations using the SW-420 sensor.
- Activates local alerts via LED and buzzer.
- Sends emergency notification to Telegram through Wi-Fi.

13.5 Assembly & Setup

Hardware Setup

- Connect the vibration sensor to pin D0 of ESP8266.
- Connect the buzzer to D8 and LED (via resistor) to D4.
- Power the ESP8266 via USB or regulated 5V adapter.
- Use a breadboard and jumper wires for prototyping.

Software Setup

1. Install **Arduino IDE** and the **ESP8266 board package**.
2. Open the provided Arduino code.
3. Replace:
    - ssid and password with your Wi-Fi credentials
    - botToken and chatID with your Telegram bot details
4. Upload the code to ESP8266 using a USB cable.

13.6 Telegram Bot Setup

1. Open Telegram → search **BotFather** → create a bot → get **token**.
2. Start a chat with your bot → visit this URL to find chat ID:
   https://api.telegram.org/bot<YOUR_TOKEN>/getUpdates
3. Copy the bot token and chat ID into the Arduino code.

13.7 How to Use

- Power the system via USB/adapter.
- Wait for Wi-Fi connection (check Serial Monitor).
- When vibration is detected:
    - LED and buzzer activate for 2 seconds
    - Telegram message is sent instantly
- After the vibration ends, the system auto resets.

13.8 Troubleshooting

Issue	Solution
Wi-Fi not connecting	Recheck SSID/password or router settings
No Telegram alert	Check bot token and chat ID
False vibration triggers	Use solid mounting, adjust code sensitivity
Buzzer/LED not working	Confirm wiring and pin assignments

10.3 Maintenance Tips

- Secure the sensor properly to avoid false readings.
- Regularly test the system for reliability.
- Keep firmware up to date.
- Store in a dry, stable environment.

13.10 Support

For assistance, contact the developer team:

Email: asraful@students.diu.ac rased@students.diu.ac

GitHub: https://github.com/asrafulmolla/Earthquake-Detector-Alarm-and-Send-Notification