THE SHAKELESS 60

LoRa-Based Earthquake Early Warning System for Urban and Remote Communities

Low-Cost Seismic Detection & Mobile Alert Prototype

A Smart Disaster Response System for Africa and Beyond

Prepared By:
Abigail Mekonnen and Beshayer Mohammedtaj
Department of Electromechanical Engineering
Addis Ababa Science and Technology University
Addis Ababa, Ethiopia

June, 2025

Abstract

This paper describes development of an affordable, LoRa-based earthquake early warning system to provide early warning before the occurrence of injurious seismic waves, primarily S-waves. The system uses a piezoelectric sensor and a MEMS accelerometer to sense ground vibration with high sensitivity. The sensors are connected to a microcontroller (Arduino Uno), which processes the data and transmits vital vibration reading through a LoRa (RA-02) module. At the receiving end, another microcontroller accumulates data transmitted and, on observing significant seismic activity, triggers a warning by means of a GSM module to provide SMS alerts. Use of long-distance, low-power LoRa communication makes it possible to send notifications even to remote or infrastructure-less regions. The system is tested using simulated vibrations from a shaker table to replicating true earthquake events, and the thresholds of the data are being set to distinguish between dangerous and harmless tremors. With the promise of offering a live, wireless, and low-cost solution, this prototype holds the potential to help in disaster preparedness and public safety, especially in seismically active areas like the African Rift Valley. As one of the most common earthquake prone locations in Africa, we used Ethiopia's Afar region for demonstration purposes.

TABLE OF CONTENTS

1. Identification Statement of the Problem

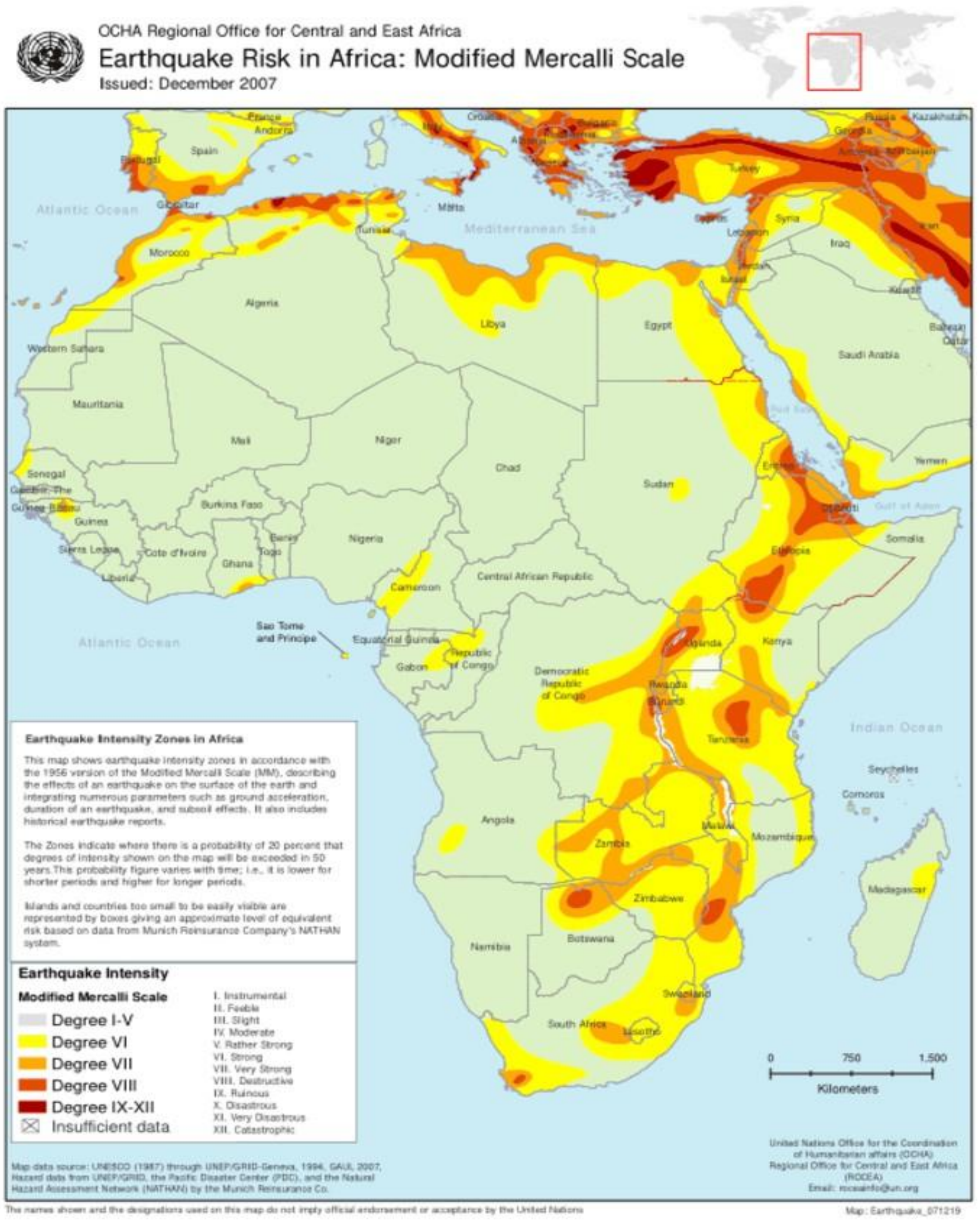
The East African Rift Valley is a dynamic tectonic zone where the African Plate is gradually splitting into two—a geological process that manifests in regular earthquake activity. In 2024–25, the region around Ethiopia's Afar and Oromia underwent a swarm of over 300 earthquakes, with large tremors at 5.9 Mw in February 2025 and 5.7 Mw in January 2025. These events underscore Ethiopia's escalating seismic hazard, which is a concern due to the exposure of its rural and urban populations.

Worldwide, the September 8, 2023, Morocco earthquake—a 6.9 Mw (initially reported as 6.8 Mw) oblique-thrust earthquake near the vicinity of Al Haouz in the Atlas Mountains—killed nearly 3,000 individuals, wounded thousands of others, and caused extensive damage to historic infrastructure. Even though Morocco lies outside the eastern arm of the Rift Valley, its disastrous earthquake is a grim reminder: significant seismic activity can occur at unexpected locations, typically with catastrophic result.

Despite the frequent occurrence of seismic activity in the Horn of Africa, an extensive, low-cost, and decentralized early warning system does not yet exist in Ethiopia. Existing seismic monitoring infrastructure is either low in coverage, relies on costly equipment, or requires internet connectivity—constraints that limit their application in remote or resource-scarce regions.

The rapid onset of destructive S-waves—following P-waves—means that citizens in impacted regions typically receive little or no warning before damage arrival, increasing casualties and endangering critical infrastructure. The recent Ethiopian earthquakes (5.2–5.9 Mw swarm events) bear witness to the urgency of localized warning systems. Further, the ongoing rift splitting in Eastern Africa suggests that future seismicity is likely, and advance safety measures are even more warranted.

This project aims to fill this gap by developing a low-cost LoRa-based earthquake early warning and detection system with the capability to deliver mobile phone warnings up to one minute before S-wave arrival, enhancing preparedness and reducing hazard for populations across Ethiopia and other analogous tectonic regions.



Earthquake Intensity Zones in Africa

This map shows earthquake intensity zones in accordance with the 1956 version of the Modified Mercalli Scale (MM), describing the effects of an earthquake on the surface of the earth and integrating numerous parameters such as ground acceleration, duration of an earthquake, and subsoil effects. It also includes historical earthquake reports.

The Zones indicate where there is a probability of 20 percent that degrees of intensity shown on the map will be exceeded in 50 years. This probability figure varies with time; i.e., it is lower for shorter periods and higher for longer periods.

Islands and countries too small to be easily visible are represented by boxes giving an approximate level of equivalent risk based on data from Munich Reinsurance Company's NATHAN system.

Map data source: UNESCO (1987) through UNEP/GRID-Geneva, 1994, GAUL 2007, Hazard data from UNEP/GRID, the Pacific Disaster Center (PDC), and the Natural Hazard Assessment Network (NATHAN) by the Munich Reinsurance Co.

United Nations Office for the Coordination of Humanitarian affairs (OCHA)
Regional Office for Central and East Africa (ROCEA)
Email: rocea-info@un.org

The names shown and the designations used on this map do not imply official endorsement or acceptance by the United Nations

Map: Earthquake_071219

Source: https://reliefweb.int/map/ethiopia/earthquake-risk-africa-modified-mercalli-scaledecember-2007

2. Scope of the Project

The project seeks to design, develop, and test a low-cost LoRa-based earthquake early warning system (EEWS) to be employed for detecting seismic movement and providing real-time warnings to communities. It will be designed to be applied in urban and rural areas of seismic risk, including the East African Rift Valley with particular application to Ethiopia's geologically active Afar and Oromia regions.

The project involves the utilization of piezoelectric sensors and accelerometers and a microcontroller (Arduino Nano) to detect ground vibration. Such signals are transmitted by LoRa (Long Range) wireless communication to a receiver node, which evaluates vibration data. In case seismic activity with amplitude higher than a given threshold is detected, an SMS message is triggered to targeted users through a GSM module.

The work includes:

1. Hardware implementation of sensor and gateway nodes.
2. Software implementation of sensor data acquisition, LoRa communication, and GSM alerting.
3. Calibration of thresholds of vibrations to discriminate between false alarms and real seismic events.
4. Tests and validations using controlled vibration sources such as shaker tables or bass transducers. System performance tests for short to medium range distances in open field environments.

This prototype is scalable and may be scaled up to a network of sensor nodes for larger regional coverage. The system is low power, offline-capable (no internet), and fast deployment, which makes it suitable for low-resource environments.

3. Literature Review

Recent advances in earthquake early warning systems (EEWS) have demonstrated that low-cost, longdistance communication devices like LoRa can be highly effective in online monitoring of earthquakes, especially in regions with poor infrastructure. Researchers have demonstrated that MEMS-based sensors like piezoelectric material and accelerometers can detect seismic activity efficiently when coupled with proper amplification and filtering. Most significantly, implementations like QuakeSense have established LoRa's capability of delivering seismic signals over long distances with minimal latency, making multihop LoRa setups able to deliver alerts in seconds over up to 30 km—comparable with traditional internetbased systems. These findings strongly support the potential and reliability of our new earthquake detection system based on LoRa, aimed at alerting city networks one minute before the onset of destructive S-waves. In addition, advancements like ADC ADS1115 chips have been explored to achieve maximum signal resolution and effectively sense low-frequency vibrations. Overall, this work proves that an EEWS like ours that is scalable, low-power, and low-cost can play a vital role in safeguarding communities through mobile warnings and decentralized sensing.

4. Procedure (Methodology)

The project involves the design and implementation of a low-cost, LoRa-based earthquake early warning system (EEWS) which detects seismicity using ground vibration sensors and disseminates real-time warnings to the general public before the arrival of damaging S-waves.

4.1. System Architecture

The system has two main components:

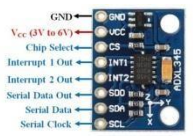
a) A Sensor Node (Sender Side) which detects ground vibrations using a piezoelectric sensor and accelerometer. Pre-processing is done with an op-amp prior to transmission via a LoRa RA-02 module.

b) A Gateway Node (Receiver Side) receiving data from the sender via LoRa. If seismic activity level beyond a specified threshold is observed, an alert is raised and sent via SMS with the help of a GSM module.

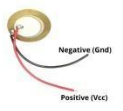
4.2. Hardware Setup

4.2.1 Sensor Node (Sender)

A piezoelectric sensor is connected to an op-amp circuit to amplify signals. The Amplified and filtered analog signals are detected by an Arduino Uno. An accelerometer (ADXL345) is integrated for better accuracy. The Data arrives via LoRa (433 MHz) via the RA-02 module.



Accelrometer



Piezoelectric Sensor

LoRa Module (Sender)

Arduino Uno

4.2.2. Gateway Node (Receiver)

LoRa Module (Reciever)

GSM Module

Arduino Uno

Another Arduino Uno receives incoming LoRa packets.

When vibration data exceeds a pre-set threshold, the Arduino triggers an alarm through a GSM module . The alarm is triggered as an alert SMS to pre-programmed numbers.

4.3. Communication Protocol

Although this deals with wireless transmission, as LoRa is involved, which consumes low power and does not need internet for operation, it has been included here as wireless communication. Communication should be one-way: sensor node to gateway.

Detailed Design and Schematics

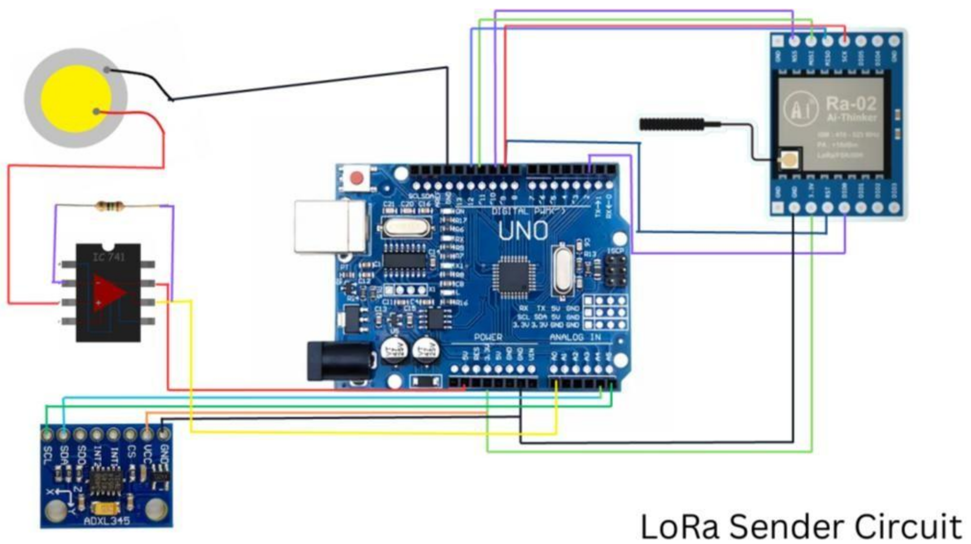
1. Sensor Node (Sender Side)

Components:

- Arduino Uno (Microcontrollers)
- Piezoelectric Sensor (for seismic activity detection)
- Accelerometer (ADXL345 for 3-axis motion detection)
- Operational Amplifier (LM358) to amplify sensor signals
- LoRa RA-02 Modules (for wireless transmission)
- Resistors and Capacitors (for filtering and stabilization)
- PCB board (for prototyping)
- Jumper Wires
- Power Supply (USB from PC or 5V battery pack) Vibration Motor (to simulate the earthquake)

Wiring & Setup:

1. Connect the piezoelectric sensor to the op-amp circuit and then to an analog input pin on the Arduino.
2. Connect the accelerometer via I2C (SDA to A4, SCL to A5 on Arduino Nano).
3. Connect LoRa RA-02 : MOSI to D11, MISO to D12, SCK to D13, NSS to D10.
4. Power the Arduino via USB or a regulated 5V source.
5. Use software filtering on Arduino to clean and package sensor data.
6. Transmit the data over LoRa to the gateway (receiver side).



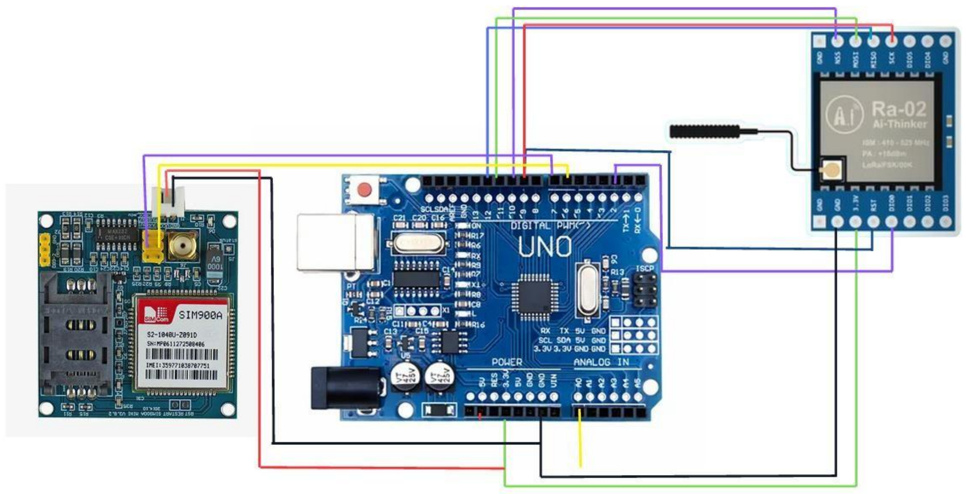
LoRa Sender Circuit

2. Gateway Node (Receiver Side)
This node receives seismic data over LoRa, processes it, and sends alerts via GSM or displays it on a PC/server.

Components:
- Arduino Nano (or similar Microcontroller)
- LoRa RA-02 Module (for receiving data)
- GSM Module (SIM800L for sending SMS alerts)
- Resistors and Capacitors (as needed for stable communication)
- Jumper Wires and Breadboard Power Supply (USB or external 5V)

Wiring & Setup:

1. Connect the LoRa RA-02 to the Arduino using the SPI pins (MOSI, MISO, SCK, NSS).
2. Connect the GSM module to the Arduino: TX of GSM to RX of Arduino, and RX of GSM to TX of Arduino.
3. Power the GSM module using a dedicated 5V/2A supply or powered USB hub.
4. Program the Arduino to decode incoming LoRa packets and trigger SMS alerts if a threshold is met.
5. Connect the Arduino to a PC to log and analyze seismic activity.



LoRa Receiver Circuit

Other Components for Full System Deployment:

- Vibration motor (for earthquake simulation)
- Local Server or a PC (for advanced data analysis)

Summary of the Connections for the System

Piezoelectric Sensor and Op-Amp Circuit

1. Connect the positive terminal of the piezo sensor to the non-inverting input (pin 3) of the op-amp.
2. Connect the negative terminal of the piezo sensor to GND.
3. Connect the output (pin 6) of the op-amp to the analog input A0 of the Arduino Uno.
4. Connect a resistor between the output (pin 6) and the inverting input (pin 2) of the op-amp.
5. Connect another resistor between pin 2 and GND.
6. Add a capacitor between the op-amp output (pin 6) and GND to filter high-frequency noise.
7. Power the op-amp: connect V+ (pin 7) to 5V, and V− (pin 4) to GND.

LoRa Module Connections

1. VCC → 3.3V on Arduino Uno
2. GND → GND
3. SCK → D13
4. MISO → D12
5. MOSI → D11
6. NSS → D10

7. RST → D9
8. DIO0 → D2

Accelerometer Connections

1. VCC → 3.3V on Arduino Uno
2. GND → GND
3. SDA → A4
4. SCL → A

4.4 Software Implementation

Microcontrollers are coded using Arduino IDE. The sender sketch reads analog values from the piezo sensor and accelerometer, processes them, and transmits the reading if vibration is detected. The receiver sketch accepts received packets and sends an SMS alert if the signal exceeds a threshold.

Sender Arduino IDE code:

```
// LORA SENDER (SENSOR NODE)
// ADXL345 (I2C) and LoRa RA-02

#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_ADXL345_U.h>
#include <SPI.h>
#include <LoRa.h>

// This is the LoRa Configuration
#define LORA_FREQUENCY 433E6
#define SS_PIN    10
#define RST_PIN   9
#define DIO0_PIN  2
#define SERIAL_BAUD_RATE 9600

// The motion detection Threshold
float movementThreshold = 0.3; // Lowered for better sensitivity (was 0.5)
// ADXL345 Sensor Section
Adafruit_ADXL345_Unified accel = Adafruit_ADXL345_Unified(12345);
sensors_event_t prevEvent;

void setup() {
  Serial.begin(SERIAL_BAUD_RATE);   while
(!Serial); // Wait for Serial Monitor
  Serial.println("--- LoRa & ADXL345 Sender Initializing ---");
```

```
  // The following is the initialization of
ADXL345   Serial.println("Initializing
ADXL345...");   if (!accel.begin()) {
    Serial.println("❌ ADXL345 not found. Check connections!");
while (1);
  }
  Serial.println("❌ ADXL345 Ready.");
accel.setRange(ADXL345_RANGE_16_G);
accel.getEvent(&prevEvent); // Store initial reading

  // Initializing LoRa
  Serial.println("Initializing LoRa...");
LoRa.setPins(SS_PIN, RST_PIN, DIO0_PIN);   if
(!LoRa.begin(LORA_FREQUENCY)) {
    Serial.println("❌ LoRa init failed. Check wiring/power.");
while (1);
  }
  LoRa.setSpreadingFactor(10);
  LoRa.setSignalBandwidth(125E3);
  LoRa.setCodingRate4(5);
  LoRa.setPreambleLength(8);
  LoRa.setSyncWord(0x12);
  Serial.println("❌ LoRa Ready. Waiting for motion...");
  Serial.println("------------------------------------");
}  void loop() {
sensors_event_t event;
accel.getEvent(&event);

  // Change in acceleration   float dx = abs(event.acceleration.x
- prevEvent.acceleration.x);   float dy =
abs(event.acceleration.y - prevEvent.acceleration.y);   float dz
= abs(event.acceleration.z - prevEvent.acceleration.z);
  // Update previous reading every time
(critical!)   prevEvent = event;
  // Motion detected
  if (dx > movementThreshold || dy > movementThreshold || dz > movementThreshold)
{
    String message = "X: " + String(event.acceleration.x,
2);     message += " Y: " + String(event.acceleration.y, 2);
message += " Z: " + String(event.acceleration.z, 2);
```

```
    Serial.println("❌ Sending: " + message);
    LoRa.beginPacket();
    LoRa.print(message);
    LoRa.endPacket();
  }    delay(200); // A Small delay is put to control speed of
sending(~5 Hz)
}
```

Receiver Arduino IDE code:

```
#include <SPI.h>
#include <LoRa.h>
#include <SoftwareSerial.h>

// Recevier LoRa Configuration
#define SS_PIN    10
#define RST_PIN   9
#define DIO0_PIN  2
#define LORA_FREQUENCY 433E6
#define SERIAL_BAUD_RATE 9600

// Our GSM SIM Module
#define GSM_RX 6  // GSM TX → Arduino RX
#define GSM_TX 7  // GSM RX ← Arduino TX
SoftwareSerial gsmSerial(GSM_TX, GSM_RX); // TX, RX


//These are the SMS details_ the phone number can be here String
phoneNumber = "+2519XXXXXXXX";
String alertMessage = "Earthquake alert in one minute!!";

void setup() {
  // Serial Monitor
  Serial.begin(SERIAL_BAUD_RATE);
while (!Serial);

  // LED Setup
pinMode(TRIGGER_LED, OUTPUT);
digitalWrite(TRIGGER_LED, LOW);
```

```
 // GSM Init   gsmSerial.begin(9600);
delay(1000);   gsmSerial.println("AT");
delay(1000);   gsmSerial.println("AT+CMGF=1"); //
SMS text mode   delay(1000);
gsmSerial.println("AT+CLIP=1"); // Caller ID
delay(1000);

  Serial.println("❌ GSM ready.");

  // LoRa Init
  LoRa.setPins(SS_PIN, RST_PIN, DIO0_PIN);
if (!LoRa.begin(LORA_FREQUENCY)) {
    Serial.println("❌ LoRa init failed. Check wiring/power.");
while (1);
  }

  // Match sender's settings
  LoRa.setSpreadingFactor(10);
  LoRa.setSignalBandwidth(125E3);
  LoRa.setCodingRate4(5);
  LoRa.setPreambleLength(8);
  LoRa.setSyncWord(0x12);

  Serial.println("LoRa Receiver + GSM is Ready.");
}  void loop() {   int packetSize =
LoRa.parsePacket();

  if (packetSize > 0) {
    String incoming = "";
     while (LoRa.available()) {
incoming += (char)LoRa.read();
    }
    incoming.trim(); // Clean input
    Serial.print("❌ Received LoRa message: ");
    Serial.println(incoming);
```

```
void sendSMS() {
  Serial.println("❌ Sending SMS alert...");

gsmSerial.println("AT+CMGF=1");
delay(500);

gsmSerial.print("AT+CMGS=\"");
gsmSerial.print(phoneNumber);
gsmSerial.println("\"");
delay(5000);

gsmSerial.print(alertMessage);
delay(500);

  gsmSerial.write(26); // Ctrl+Z to send
  Serial.println("❌ SMS sent to " + phoneNumber);
}
```

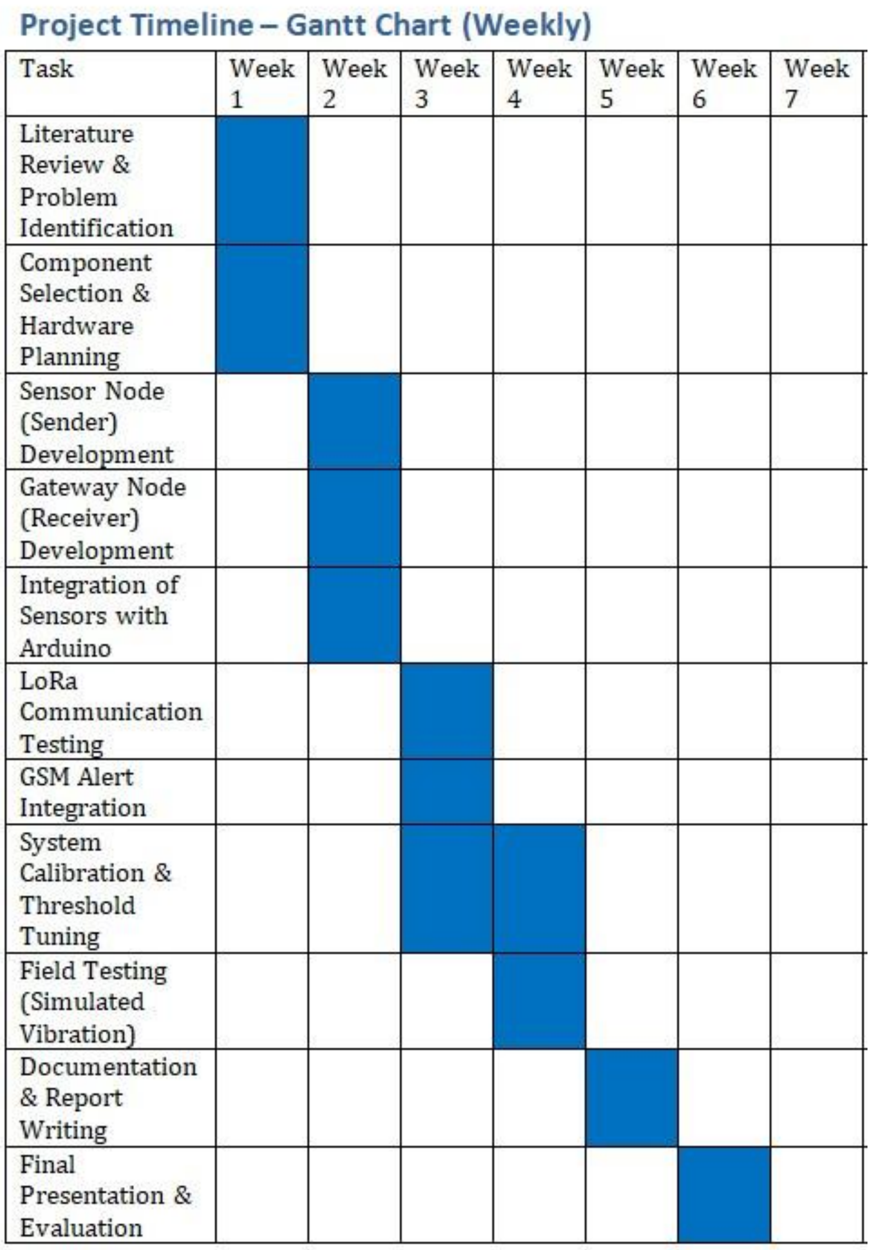
4.5 Testing and Calibration

The vibration motor is used to mimic S-waves. A number of thresholds were tested to reach the best sensitivity and reduce false positives.

Data transmission was verified in short- and medium-distance. The results are excellent in areas with no barriers as the LoRa transmission is most effective in open places as such. (a LoRa can cover up to **15km** range in open coverages). SMS delay of delivery was tracked so the alert reaches in 10 seconds of detection.

Deployment Consideration : Sensors are buried 1–5 meters deep in solid ground (volcanic areas, like Afar region). LoRa nodes will be powered by USB for prototype development; subsequent versions will use solar panels. Gateway may be connected to a PC as a local server for the prototype for data sophisticating logging and analysis.

5. Time-Line of the Project



Project Timeline – Gantt Chart (Weekly)

Task	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7
Literature Review & Problem Identification	🟦						
Component Selection & Hardware Planning	🟦						
Sensor Node (Sender) Development		🟦					
Gateway Node (Receiver) Development		🟦					
Integration of Sensors with Arduino		🟦					
LoRa Communication Testing			🟦				
GSM Alert Integration			🟦				
System Calibration & Threshold Tuning			🟦	🟦			
Field Testing (Simulated Vibration)				🟦			
Documentation & Report Writing					🟦		
Final Presentation & Evaluation						🟦	

6. **Detailed Explanation of the Transmission**

The system detects seismic activity using sensors, processes data on a local server and sends SMS alerts to resisdents is the S waves are detected.

A. Sensor and Gateway Setup

The piezoelectric sensors are connected to the LoRa system using a microcontroller. These sensors generate small voltage signals when they detect vibrations in the ground. The microcontroller reads these signals through its analog input pins and processes them to remove noise and keep only the important information, like the strength and frequency of the vibrations. After processing, the microcontroller sends this data to the LoRa module.

The LoRa module converts the data into packets and sends them wirelessly to the LoRa node. The LoRa node, which includes both the microcontroller and the LoRa module, ensures that the vibration signals are transmitted over long distances to the LoRa gateway. This setup allows the sensors to effectively and reliably send important seismic information to the gateway for further analysis.

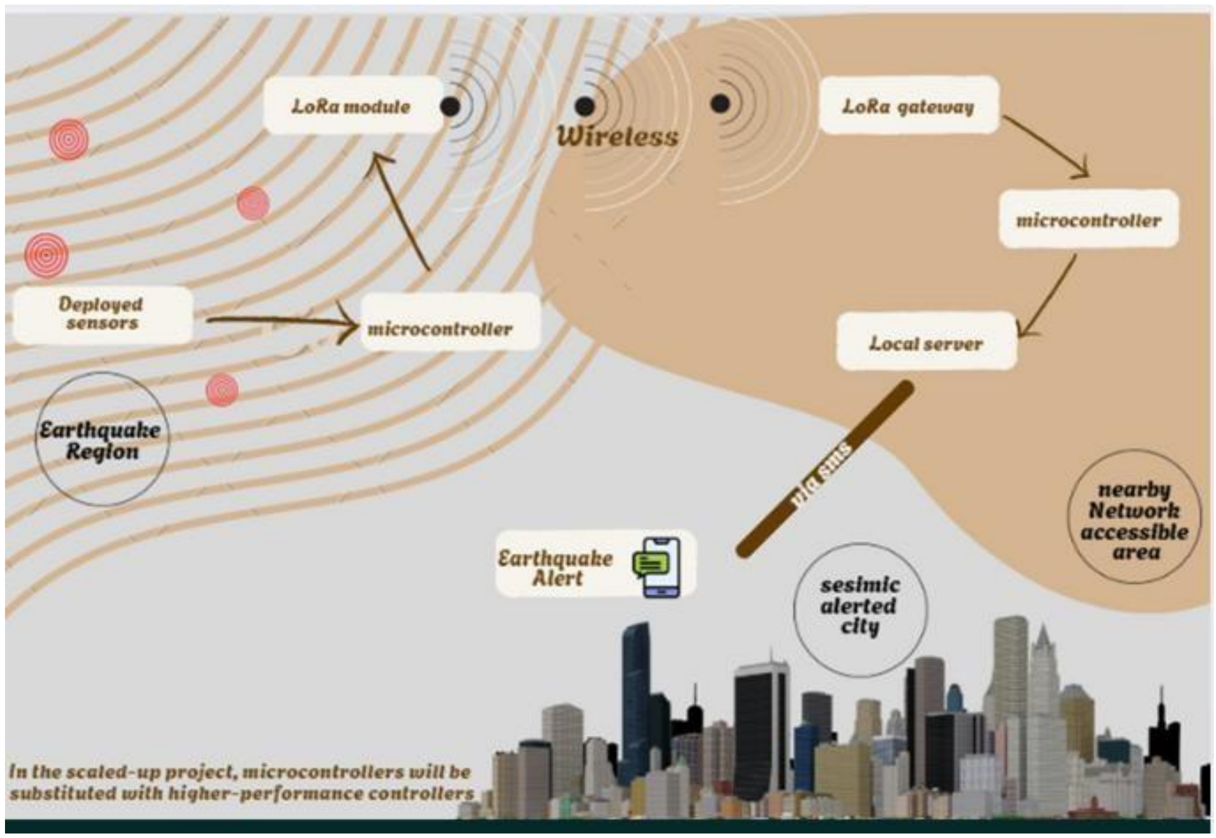
B. Placement/deployment:

Since we're placing the piezoelectric sensor in areas (Afar region) with solid volcanic rock (like basalt or compacted soil), it would likely perform well because seismic waves, including S-waves, travel efficiently through solid rock. If the sensor is placed in solid volcanic rock or compacted soil, it should work relatively well. These areas can transmit seismic waves effectively, and we won't need to worry about excessive signal loss. A depth of 1–5 meters would still be effective in detecting S-waves in this type of terrain.

C. Server Processing

The server (a PC for the prototype) receives seismic data from the LoRa gateway, which includes information such as amplitude, frequency, and timestamp. The LoRa gateway is connected to a second microcontroller, which manages communication with the LoRa module and forwards the data to the server. Upon receiving this data, the server processes it by applying a band-pass filter to isolate seismic frequencies within the range of 0.1 to 10 Hz, effectively removing unwanted noise. Once filtered, the server analyzes the data to distinguish between P-waves and S-waves. P-waves, which have smaller amplitudes and higher frequencies, are ignored as they do not pose a direct threat. However, if an S-wave is detected, identified by its larger amplitude and lower frequency, the server confirms it as a seismic threat. When an S-wave is detected, the server quickly calculates the time remaining before it reaches the target area. It then sends an SMS alert using a connected GSM module. The alert message is transmitted via mobile to residents or local authorities.

Below is a diagram illustrating the workflow and the placement of the devices. It provides a clear overview of the system's operation and device arrangement.



Calculating How the Alert Precedes the S-Wave before reaching cities

Key Distances and Speeds:

- Distance from earthquake epicenter to target city (e.g Addis Ababa): 600 km.
- P-wave speed (v_p)6 km/s.
- S-wave speed (v_s): 3.5 km/s.

Key Times:

- P-Wave Arrival Time (T_p):

$$T_p = \frac{Dis\tan ce}{} = \frac{600}{} = 100\ \text{sec(1min 40 sec)}$$

$$V_P \qquad 6$$

- S-Wave Arrival Time (T_s):

$$T_s = \frac{Distance}{V_s} = \frac{600}{3.5} \approx 171\text{sec}(2 \text{ min } 51\text{sec})$$

- Warning Time (T_w):

$$T_w = T_s - T_p = 171 - 100 = 71 \text{ seconds } (1 \text{ min } 11 \text{ sec})$$

How the Server Handles Timing

- S-Wave Confirmation:
  - Time for S-wave detection and alert sending:

$$T_a = \mathbf{1 - 3 \text{ seconds}}$$

- Gateway to Server**:**

  The gateway transmits the P-wave data to the server.

  Time taken for transmission: ~1 second.

  SMS transmission time (~10 seconds)

- Remaining Warning Time:
  71 sec - (1 sec + 10 sec) = 60 sec (1 min)
  - The alert reaches the city well before the S-wave arrives, giving 1 min of warning time. ***Thus, the residents can have a "shakeless 60" seconds before disaster strikes!***

Addressing LoRa's Connectivity

LoRa technology is highly suitable for the Afar region due to its ability to transmit data over long distances without relying on internet connectivity. The system uses two LoRa devices: one LoRa module connected to the sensors (LoRa node) and another at the gateway (LoRa gateway). The LoRa node, attached to sensors like piezoelectric or acoustic detectors, collects seismic data and transmits it wirelessly to the LoRa gateway. The LoRa gateway, positioned centrally, receives this data and forwards it to the local server via wired connections like USB or Ethernet. This communication occurs entirely over

LoRa's low-power radio frequencies, enabling seamless data transfer without requiring cellular or Wi-Fi networks. This setup ensures reliable and efficient real-time communication in remote regions like Afar, where internet infrastructure is limited or unavailable.

7. Conclusion

This project is a proof of concept for a low-cost and functional LoRa-based earthquake early warning system that has the ability to detect seismic motion and issue real-time warnings. By integrating a piezoelectric sensor and accelerometer with a LoRa communications module and GSM warning system, the prototype can successfully monitor ground movement and transmit warning signals in advance of the arrival of the destructive S-waves. The system is operable without internet connectivity, making it especially valuable for under-connected or rural regions such as the East African Rift Valley. The system proves itself as a scalable and community-level disaster preparedness system through testing and calibration. In total, this project contributes to the improvement of early warning infrastructure in seismically vulnerable regions, assisting in the greater goal of saving lives and minimizing damage by using proactive technological intervention.

8. **Future Improvements**

While the current prototype convincingly demonstrates the viability of a low-cost, LoRa-based earthquake early warning system, several upgradations can significantly boost its reliability, coverage, and ease of use in the future:

Multi-Sensor Integration:

Add more seismic sensors (e.g., geophones or professional MEMS sensors) to improve detection accuracy and suppress false alarms, especially in noisy urban environments.

Real-Time Networked System:

Add an array of sensor nodes deployed in different geographic locations to enable triangulation and faster confirmation of seismic activity, and improve the dependability and responsiveness of the system.

Machine Learning-Based Filtering:

Utilize machine learning methods to filter out the seismic signals and distinguish authentic earthquakes from sources of environmental noise (e.g., movement of humans, car traffic, machinery).
Develop a mobile application that not only sends alerts but also shows real-time maps of seismic activity, emergency contacts, and safety protocols for users.

Extended Power Autonomy:

Use energy-efficient equipment and combine solar-powered battery systems to facilitate long-term use in off-grid areas without regular maintenance.

Field Deployment & Validation:

Collaborate with national geoscience and meteorological authorities to deploy and test the system in real seismic areas such as Afar and the Rift Valley, collecting real data for improved performance.

9. Summary

The local server, placed near the LoRa gateway, processes seismic data quickly after receiving it. When a Pwave is detected, the server calculates the time remaining for the S-wave to reach the target area. For example, if the S-wave takes approximately 171 seconds to cover 600 km, and the P-wave reaches the sensors in 100 seconds, the available warning time is 71 seconds. Note that this assumption is taken as if a P wave and an S wave emerge at the same time, but in **reality there is a time gap between them**, giving the system more seconds for the alert. The assumption is just for calculation purpose. After subtracting processing time (~1 seconds) and SMS transmission time (~10 seconds), the server ensures the alert is sent with at least 60 seconds remaining. This allows residents to prepare before the S-wave arrives.

10. References

1. Algiriyage, H., et al. (2022). Earthquake early warning systems based on low-cost ground motion sensors: A systematic literature review. Frontiers in Sensors.

2. Boccadoro, P., Montaruli, B., & Grieco, L. A. (2019). QuakeSense, a LoRa-compliant Earthquake Monitoring Open System. DS-RT IEEE/ACM.

3. [Anonymous]. (2024). IoT-Based Wireless Networking for Seismic Applications. ―Moonlight Review‖.