BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

EEE 416

Microprocessor and Embedded Systems Laboratory

Final Project Report

Section: A2 Group: 06

Remote Earthquake Alert System with IoT

Course Instructors:

Dr. Zunaid Baten, Associate Professor, Department of EEE, BUET
Bejoy Sikder, Lecturer, Department of EEE, BUET

Signature of Instructor: ____

Academic Honesty Statement:

IMPORTANT! Please carefully read and sign the Academic Honesty Statement, below. Type the student ID and name, and put your signature. *You will not receive credit for this project experiment unless this statement is signed in the presence of your lab instructor.*

"In signing this statement, We hereby certify that the work on this project is our own and that we have not copied the work of any other students (past or present), and cited all relevant sources while completing this project. We understand that if we fail to honor this agreement, We will each receive a score of ZERO for this project and be subject to failure of this course."

Signature: ________________ **Full Name: Tanvir Hossain Arnob** **Student ID: 1906040**	**Signature:** ________________ **Full Name: Tasmin Khan** **Student ID: 1906055**
Signature: ________________ **Full Name: Md Hasib Ur Rashid** **Student ID: 1906059**	**Signature:** ________________ **Full Name: A.K.M. Anindya Alam** **Student ID: 1906065**

Table of Contents

1 Abstract

This project aims to develop a cost-effective and accurate remote earthquake detection and alert system using Internet of Things (IoT). The device developed in this project uses MPU 6050 as its primary sensor, and analyses the accelerometer and gyroscope readings to accurately detect the primary wave associated with earthquake. After detection, it sends an alert to the users, giving them some headroom for evacuation before the secondary wave can cause severe damage. It also automatically cuts off loads post-detection since industrial loads, electric lines, gas lines and elevators can be damaged during earthquake. After comparison of two different sensors- ADXL and MPU 6050, the latter one was used since it gave more accurate outputs. The project also stores the earthquake data in a local SD card and simultaneously uploads it in an IoT platform for further analysis of earthquake data. Since Bangladesh lies in an earthquake risky zone, a cost-effective earthquake detection device can be useful in the socioeconomic conditions of our country.

2 Introduction

Since the beginning of 2023, Bangladesh has witnessed 6 incidents of earthquakes with magnitude over 5.5 [1]. Bangladesh is surrounded by a number of tectonic blocks responsible for many earthquakes in the past. Calcutta, Assam and Tripura are three very earthquake prone zones, which are close to Bangladesh, making the region very vulnerable to earthquakes. [2] Earthquake devices like Seisomographs or Seisomometers or other modern earthquake detection devices are usually very expensive and not affordable, given the socioeconomic context of Bangladesh. Moreover, most of the modern earthquake detection devices use high-end sensors for accurate detection. Google uses accelerometers in smartphones for their earthquake alert system. When multiple smartphones across a region give accelerometer readings above a certain threshold, Google identifies the event as an earthquake and alerts all of its users. However, an affordable and effective earthquake detection device can be an effective solution for users, especially those more vulnerable to losses due to earthquake.

3 Design

3.1 Problem Formulation (PO(b))

The project requires the use of accelerometer data for detection of earthquake. Features from Seismic events which can help in earthquake detection needed to be identified for detection procedure. In case of seismic events, two waves are involved- primary wave (P-Wave) and secondary wave (S-Wave), which are shown in figure 1.

The P-wave is a longitudinal wave with lower vertical amplitude and doesn’t disturb the medium. It usually reaches 30 seconds before the S-wave. S-wave is transverse in nature and is the primary reason of damages due to earthquake. The earthquake detection system requires the detection of P-Wave. Accurate detection of P-Wave might give the users a headroom of around 30 seconds and save precious lives.

However, the amplitude of P-wave is very low, which makes it hard to detect. The accuracy of detection thus primarily depends on the accuracy of the sensors. The received data from the accelerometers need to be processed and compared with a precise threshold for accurate detection. Moreover, disconnection of loads which are vulnerable to earthquake (Gas and electricity lines, elevators etc.) after the detection can limit damages to a greater extent.

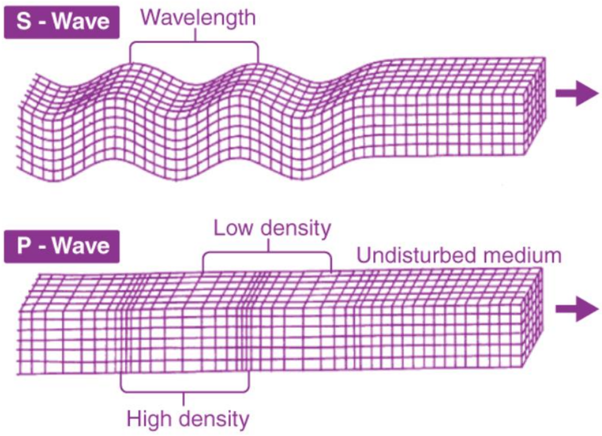


Figure 1: P wave and S wave

3.1.1 Identification of Scope

Conventional earthquake detection systems built in different projects rely on low-end sensors like ADXL, which might give inaccurate results due to the inherent noise of the sensors. Moreover, lack of integration of IoT gives no scope for detection to its users and lacks scope as an alert system. Data from a previous earthquake can also be analyzed post-earthquake for helping in future detection. Load disconnection during earthquake events can also be a useful feature for such remote earthquake detection systems.

3.1.2 Literature Review

The project requires extensive literature review, primarily for extracting features from seismic events useful for earthquake detection. A low-cost earthquake alert device using a low-cost accelerometer was implemented by Kim Et al. [5]. For understanding seismic events and their characteristics, the seismic behavior of RCC buildings under revised seismic zone classification in Bangladesh were studied, as mentioned by Ahmed, Gopal Et al. [4] For study of calibration techniques, Dorveaux, Vissiere Et al. [3] mentioned an iterative method for three-axis sensor calibration. Moreover, extensive study of the sensor datasheet were needed as well.

3.1.3 Analysis

Literature study available on early earthquake detection suggests utilization of P-Wave for earthquake detection systems. Since P-Wave is a longitudinal wave, it has lower accelerometer and gyroscope values. Different types of thresholds are used in earthquake detection systems, usually lying in between 0.01g to 0.02g (g= 9.8 m/s^2). However, a single spike above the threshold gives room for false detections. A proposed solution can be to identify the event as earthquake after certain repetitions of threshold crossing, which will reduce false detection. Moreover, post-detection earthquake data can also be used for further analysis and accurate threshold identification for increased accuracy in earthquake detection.

3.2 Design Method (PO(a))

Data acquisition and processing:

The accelerometers give data in three axes- X, Y and Z. These data need to be processed and compared with a threshold for earthquake detection. In this project, we used a threshold of 0.02g. This required continuous reading of the accelerometer data and comparing with a threshold continuously. We used two crossings of the threshold as an earthquake event to reduce chances of false detection. An ESP32 along with MPU6050 and ADXL accelerometers was used.

Accelerometers:

For the project, both ADXL and MPU 6050 were considered. ADXL readings showed more noise as compared to MPU 6050. MPU6050 has a built-in ultra-low noise linear LDO voltage regulator. It also has a built-in onboard filter and gives a digital output. On the other hand, ADXL has lower noise filtering capabilities and it gives an analog output. Both ADXL and MPU6050 were used without simulating any earthquake event to get the noise data from both sensors. The figure 2 shows that ADXL showed noise up to 0.04g whereas MPU6050 showed noise up to 0.003g.

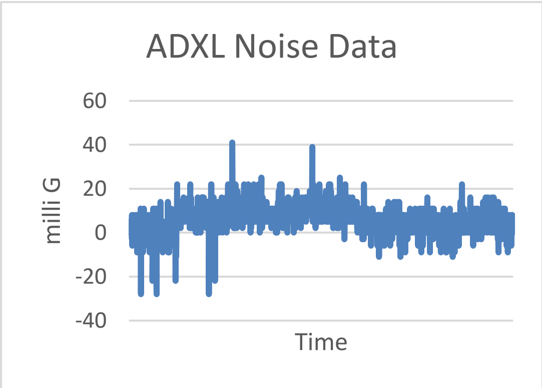


Figure 2(a): ADXL Noise Data

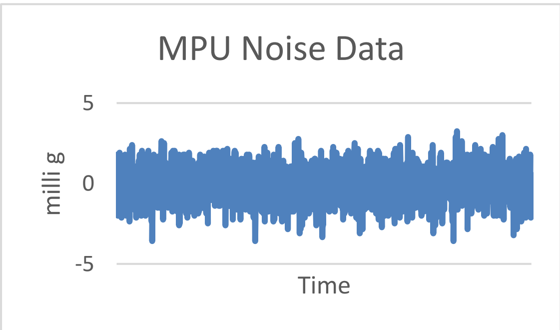


Figure 2(b): MPU 6050 Noise Data

Thus, ADXL showed less noise filtering than MPU6050 and hence, we used the latter one for earthquake detection in this project. Moreover, MPU6050 allowed the threshold to be reduced to up to 0.015g from 0.02g, which resulted in a more accurate detection.

RTC Module:

An RTC module (Precision Real Time Clock Module DS3231) was used to save the timestamp of the earthquake data so that when the alert is sent to the users, it can mention the time of the event as well. The module output showed timestamps in Hours:Minutes:Seconds format and also mentions the date as well.

Calibration:

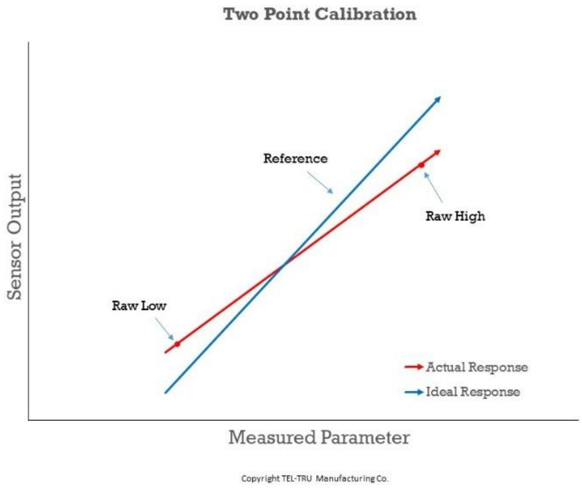


Figure 3: Two-point calibration

Two point calibration is a calibration process used to determine the accuracy of a measurement instrument or system at two distinct reference points. These reference points are usually chosen to represent the minimum and maximum expected values of measurement. After taking the two measurements, one at low end and other at high end, calculate the difference between these two values (maximum-minimum). The reference high and reference low values were also taken. Then the corrected value will be,

Corrected Value= (Raw Value- Raw Low) * (Reference Range/Raw Range) + Reference Low

In this code, the device was set at different positions to find out the raw values at each extreme (+ve X axis, -ve X axis etc.) and these values were used to calibrate the value between +1000 to -1000. +1g and -1g were used as reference high and low.

Base Offset Elimination:

Even after calibration, the device showed some small base offset which needed to be eliminated for accurate results. Average values of data for a 10 second period was taken during no-earthquake condition to find out the offset value and this offset value was subtracted from accelerometer data to find out the accurate result.

Data Storage:

For seismic events, data before and after the earthquake trigger is crucial. A SD card was used to store data locally and the same data was uploaded simultaneously to IoT platform for further analyses as well. For our project, accelerometer data of 30 seconds prior to the trigger and 1.5 minutes after the trigger was locally stored in the SD card. The data was constantly stored in the flash memory of the ESP32 and deleted after a certain interval in normal state (no earthquake). When an earthquake event was detected, the data from flash memory was transferred to the SD card. The data simultaneously sent to the IoT platform was used to plot the earthquake data. The outputs of both the sensors are shown in figure 3.

The data from both the sensors indicate less noise in case of MPU 6050, as compared to ADXL data which gives noise even during no earthquake, again proving MPU 6050 as the more suitable choice.

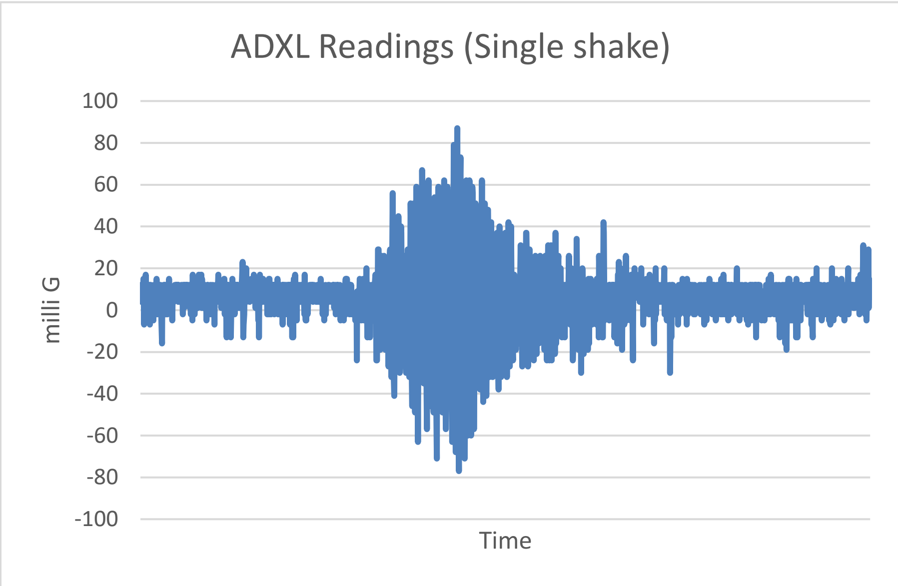


Figure 3(a): ADXL reading during earthquake

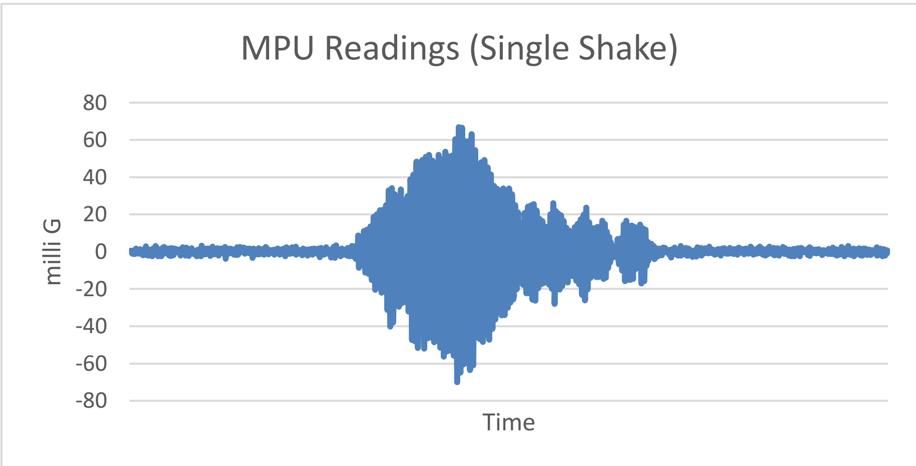


Figure 3(b): MPU 6050 readings during earthquake

Alert System and Load disconnection:

After earthquake detection, alerts were sent to the users connected to the system. Blynke IoT platform was used to send email to the users and a WhatsApp text message was also sent using another IoT platform. Moreover, the project also implemented the disconnection of remote loads using a separate ESP-32. This involved sending data from one ESP-32 to another, for which Blynke was used as well. Blynke allows communication between the two virtual devices connected to the platform (Two ESP-32s). When earthquake was detected in the primary microprocessor set in a remote location, it sent alert to the secondary microprocessor which disconnected the loads connected to that microprocessor using relay.

3.3Circuit Diagram

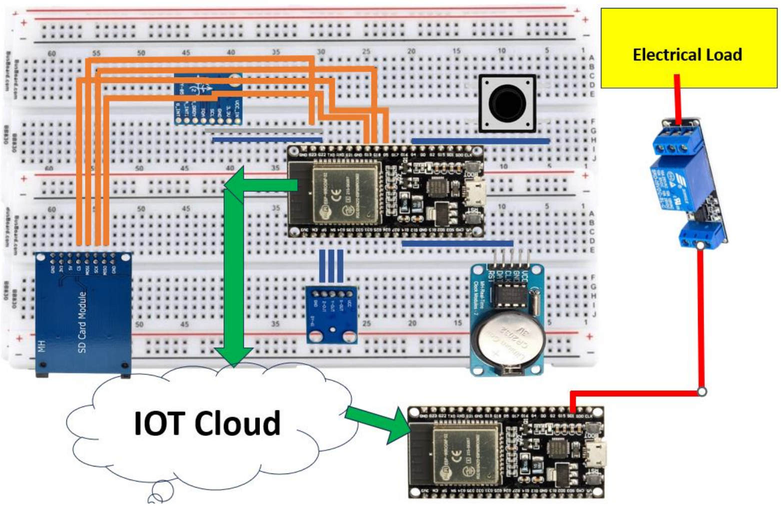
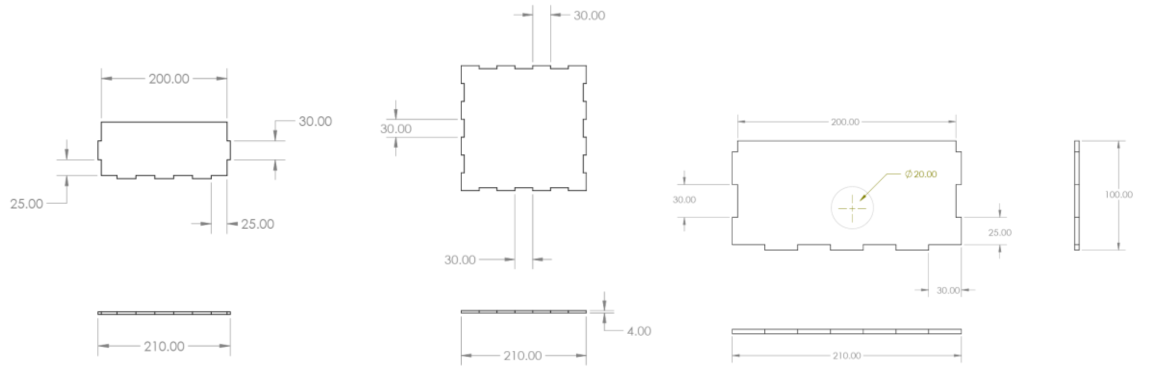


Figure 4: Circuit diagram of the earthquake detection device

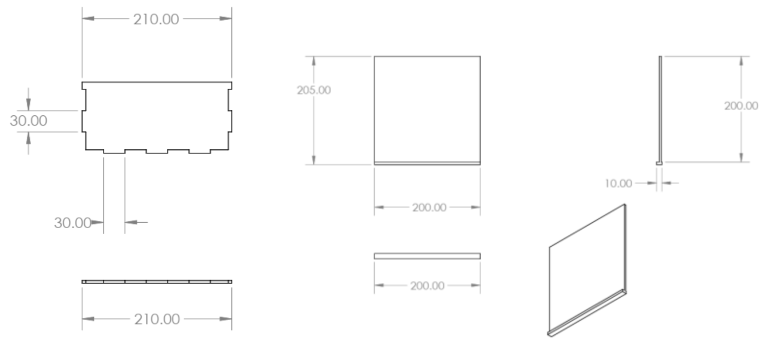
Here, two ESP-32s are being used, one for earthquake detection and the other one for electrical load disconnection. IoT is used to communicate between the two ESPs. Relay is used between the load and the second ESP for electrical communication. A RTC module is used for sending timestamps along with each data. A SD Card module along with an SD card is used for storing the data locally. Capacitors were also used with ADXL and MPU6050 for additional filtering. A pushbutton is used to re-calibrate the sensors when necessary.

3.4 CAD/Hardware Design

For testing of the device with shaker table in correspondence to Civil Engineering Department of BUET, we constructed an enclosed structure for the device. The design of the structure was done using CAD software. The structure had to be rigid enough so that the vibrations of the structure doesn't interfere with the detection of our device.



(a) Front (b) Bottom (c) Sides



(d) Rear (e) Top

Figure 5: Design of the structure in CAD

3.5 Full Source Code of Firmware

Snippets of the code is given below:

```
void data_read_all(){

  int sample = 3, i;
  float sum_xmpu = 0, sum_ympu = 0, sum_zmpu = 0;

  for(i=0; i<sample; ++i){
    sensors_event_t a, g, temp;
    mpu.getEvent(&a, &g, &temp);
    ax_mpu= -(a.acceleration.y);
    ay_mpu= a.acceleration.x;
```

This code snippet averages three consecutive data inputs to remove further noise

<pre> az_mpu= -(a.acceleration.z); sum_xmpu += ax_mpu; sum_ympu += ay_mpu; sum_zmpu += az_mpu; delayMicroseconds(20); } ax_mpu = sum_xmpu / sample; ay_mpu = sum_ympu / sample; az_mpu = sum_zmpu / sample; }</pre>	
<pre>float lerpExtend(float value, float inputMin, float inputMax, float outputMin, float outputMax) { float inputRange = inputMax - inputMin; float t = (value - inputMin) / inputRange; float result = outputMin + t * (outputMax - outputMin); return result; }</pre>	This code snippet calibrates the input digital data between a maximum and minimum value.
<pre>void set_base(int numCalib = 1000){ int i; double ax_mpuSum = 0, ay_mpuSum = 0, az_mpuSum = 0; Serial.println("...Please wait for base offset set up..."); delay(3000); for(i=0; i<numCalib; ++i){ data_read_all(); data_scale(); ax_mpuSum += ax_mpuScal; ay_mpuSum += ay_mpuScal; az_mpuSum += az_mpuScal; delay(5); } x_mpuBase = ax_mpuSum / numCalib; y_mpuBase = ay_mpuSum / numCalib; z_mpuBase = az_mpuSum / numCalib; Serial.println("Base values for mpu are x_mpuBase: " + String(x_mpuBase) + ", y_mpuBase: " + String(y_mpuBase) + ", z_mpuBase: " + String(z_mpuBase)); }</pre>	This code snippet calculates the base offset by averaging the input values for subtracting the base offset value from data inputs later on.

<pre>void data_scale(){ //value scaling:: values are expressed in milli-G unit //using map for mpu reduces its senstivity because map deals in integer long //using lerpextend for adxl makes the value stuck at a high range ax_mpuScal = lerpExtend(ax_mpu, x_mpuMin, x_mpuMax, -1000, 1000); ay_mpuScal = lerpExtend(ay_mpu, y_mpuMin, y_mpuMax, -1000, 1000); az_mpuScal = lerpExtend(az_mpu, z_mpuMin, z_mpuMax, -1000, 1000); }</pre>	This fits the input digital data between -1000 to 1000 for -g and g as maximum values
<pre>void loop() { //DateTime test1 = rtc.now(); Blynk.run(); //blynke synched timer.run(); //virtual pins updated with data every timer count if(calibrate || (!digitalRead(calButton))){ //Calibration digitalWrite(CAL,HIGH); digitalWrite (RED, LOW); digitalWrite(YEL, LOW); digitalWrite (GRE, LOW); set_base(); Serial.println("Sensor offset calibration is done"); calibrate = false; vector_mpu.clear(); start = true; Trig = false; post_rec = false; } else{ if(start){ //startTime = rtc.now(); startTime = millis(); start = false; } data_read_all(); data_scale(); ax_mpuScal -= x_mpuBase; ay_mpuScal -= y_mpuBase; az_mpuScal -= z_mpuBase;</pre>	This snippet is the main earthquake detection part of the system. After calibration, it compares the data with a certain threshold. If two consecutive events of threshold crossing occur, it sets the trigger event and saves the data to local SD card and send it to Blynk IoT platform as well. At the same time, it

<pre> //saving in the running vector vecAppend(ax_mpuScal, ay_mpuScal, az_mpuScal, vector_mpu); //comparing with threshold values if ((abs(ax_mpuScal) > xThreshold) || (abs(ay_mpuScal) > yThreshold) || (abs(az_mpuScal) > zThreshold)){ Th = true; } else{ Th = false; } if(Trig) { //DateTime current = rtc.now(); long int current = millis(); //TimeSpan elapsed = current - postRec_Time; if ((current - postRec_Time) > post_lim){ Trig = false; post_rec = false; //SD card cant save names with ":" nameStamp.replace(":", "_"); nameStamp.replace("-", "_"); SDfile_mpu(); //clear vectors vector_mpu.clear(); vector_mpu.shrink_to_fit(); start = true; pre_rec = false; warn = 0; set_base(100); } } else if(Th){ nameStamp = String(rtc.now().timestamp()); warn +=1; if(warn == 2){ date_check = 1; if (date_check == 1) { getLocalTime(); Blynk.logEvent("earthquake_event"); date_check = 0; } timer.setInterval(500, earthquakeData);</pre>	

`` data_processing(ax_mpuScal, ay_mpuScal, ax_mpuScal, 4, 5, 6); Trig = true; post_rec = true; //postRec_Time = rtc.now(); postRec_Time = millis(); } } else{ warn = 0; } } digitalWrite(CAL,LOW); if(warn == 2){ digitalWrite (RED, HIGH); digitalWrite(YEL, LOW); digitalWrite (GRE, LOW); } else if(warn>0){ digitalWrite (RED, LOW); digitalWrite(YEL, HIGH); digitalWrite (GRE, LOW); } else{ digitalWrite (RED, LOW); digitalWrite(YEL, LOW); digitalWrite (GRE, HIGH); } delay(50); } ```	

Table: Source Code for the main program

4 Implementation

4.1 Hardware

The circuit contained a capacitor with the ADXL 335 to remove any further noise offset for the sensor. It also contained an RTC module for timestamps of the earthquake events along with the data. Three LEDs were present- Red, Yellow and Green, which indicate-

Green LED- No earthquake condition
Yellow LED- One crossing of the threshold causes yellow LED to light up. Works as early detection system.
Red LED- Two crossing of the threshold and earthquake has been detected.

4.2 Load Disconnection

Two ESP-32s were used in this project- one on the detection end and another one in the load end. The detection end ESP-32 stored its accelerometer data in a local SD card and send this data on the IoT platform Blynk. Blynk is used to connect the two ESP-32. After detection of earthquake, a virtual switch in Blynk is switched off, which sends a signal to the ESP-32 on the load end, and the ESP-32 on the load end disconnects the load using a relay.

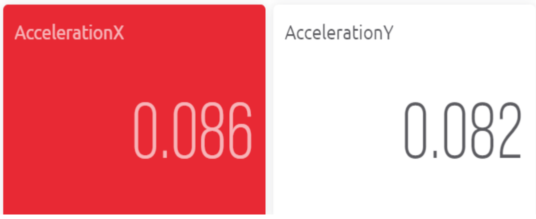


Figure 4: Accelerometer readings in Blynk platform

4.3 Alert system using IoT

Blynk is also used to send email to the concerned users when earthquake is detected. Another IoT platform named CallMeBot is used to send text messages on WhatsApp to the users when earthquake is detected.

Fig. 5: Overall circuit of the detection device

4.4 Data visualization using IoT

The data sent from the detection end ESP-32 to the IoT platform can be used to visualize the accelerometer data and use it for further analysis. ThingSpeak is an open-end IoT platform incorporated with MATLAB, which has been used in this project to visualize the earthquake data. The intensity of the data was figured out in real time by using FFT and Perseval's theorem in ThingSpeak IoT platform. The output of the ThingSpeak platform was shown in a custom app made for this project using MIT App Inventor.

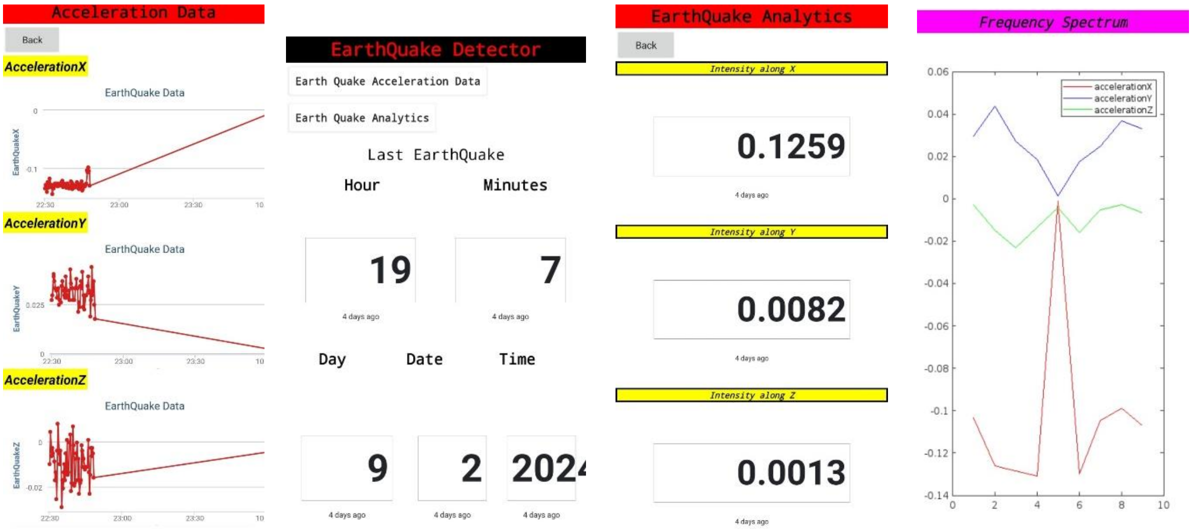


Figure 6: ThingSpeak output shown in app made using MIT App Inventor

5 Design Analysis and Evaluation

5.1 Novelty

a) **Data Storage:** Data prior to 30 seconds of the earthquake and 1.5 minutes after was saved when earthquake was detected. This data could be visualized using IoT platforms and can be used for further analyses of seismic events

b) **Calibration:** 2-point calibration was done for accurate data input

c) **Reduction of noise:** Average of 3 consecutive readings were taken to minimize the noise fluctuations. Moreover, capacitors were used to reduce noise for the sensors

d) **Double check:** The threshold needed to be crossed twice for earthquake detection. This allows for reduction of false detection owing to noise

e) **Data visualization:** Use of ThingSpeak and MIT app inventor allows the users to visualize the earthquake data along with time, accelerometer values and intensity using an app.

f) **Load disconnection:** Remote loads are disconnected using a second ESP-32, to reduce damages in industrial and other loads vulnerable to earthquake in order to minimize further damages.

5.2 Design Considerations (PO(c))

5.2.1 Considerations to public health and safety

Since Bangladesh lies in an earthquake prone zone, an effective earthquake detection and alert system could be very useful. Especially in industries, earthquake may lead to electrical failures, gas leaks and other hazards that may lead to further damages. An earthquake detection-based load-disconnection system could minimize the effects to a large extent. Keeping this in mind, the load disconnection feature was added.

5.2.2 Considerations to environment

Earthquake detection devices can also help in assessing the environmental impacts of earthquakes. They can help identify areas that are prone to landslides, liquefaction and other secondary effects of seismic events. Information received from such devices is vital for environmental management and conservation efforts. For this reason, the data of the seismic event was stored for further analysis of the seismic event.

5.2.3 Considerations to cultural and societal needs

Access to accurate and timely information of seismic activity can empower communities to develop resilience strategies. They can also contribute to public safety by providing reliable information about seismic events. This allows authorities to disseminate timely alerts and advisories. Moreover, in the aftermath of earthquake, such a detection device can play a crucial role in facilitating effective crisis management and response efforts. Keeping this in mind, the data was stored locally as well as in IoT platform to further study the seismic event.

5.3 Investigations (PO(d))

5.3.1 Design of Experiment

The two sensors- MPU6050 and ADXL335 were tested separately in both non-earthquake conditions and earthquake conditions. ADXL335 showed more noise in both cases, causing false detection alerts at times. On the other hand, MPU6050 showed more advanced performance when the accelerometer data was plotted.
Study of previous earthquake events showed that the acceleration value of seismic events is around 0.01g to 0.02g. This might be very close to any hand-stimulated vibration event. That is why, the device must be kept as a remote location away from the common vibration sources so that it can detect the earth vibration sources effectively.

5.3.2 Data Collection

After calibration, the accelerometer data showed some offset, i.e., the accelerometer was showing non-zero values even during no earthquake conditions. To resolve this matter, the average of the non-zero values were taken over the period of a few seconds and averaged, so that it could be deducted from the data to get the correct values.

5.3.3 Results and Analysis

The output data from accelerometers showed clear spikes when earthquake was simulated. Since MPU6050 showed better performance, this was used finally for our project with ADXL335 still available on the hardware end. The data was sent to the IoT platform and analyzed using ThingSpeak and accessed through the app made using MIT App Inventor.

5.3.4 Interpretation and Conclusions on Data

The data found from MPU6050 is better than ADXL335 but it still has some limitations regarding noise filtering. Moreover, the threshold of earthquake detection system is very low due to the low amplitudes of P-Wave. So, this might interfere with any man-made vibrations around us and still has chances of false detection.

5.4 Limitations of Tools (PO(e))

a) **Sensor Limitations:** The primary limitation of the project lies with that of the sensors. ADXL335 is very noise prone and MPU6050, though better than ADXL335, still has scopes of false detection. Using Piezoelectric sensors instead of MEM sensors can result in better accuracy
b) **Flash Memory Limitations:** Pre-trigger and post-trigger data saving time is limited due to low flash memory of ESP32
c) **IoT platform user limitation:** Unsubscribed use of IoT platforms results in limitations of number of users and devices that could be integrated
d) **Data transfer delay:** Unsubscribed use of IoT platforms results in limited data rate and thus causes some fair amount of delay in signal transfer
e) **Lack of testing resources:** Proper testing with a shaker table could help in developing the threshold values more accurately.

5.5 Impact Assessment (PO(f))

5.5.1 Assessment of Societal and Cultural Issues

One of the primary concerns of the early detection system would be that it should be accessible to all regardless of economic status and geographic location. Given the socioeconomic infrastructure of Bangladesh, a cost-effective earthquake device could help in more users subscribing to the system at low cost and make the device accessible to everyone.

5.5.2 Assessment of Health and Safety Issues

An early earthquake detection device, when installed in industries, could potentially save lives of its workers. One of the primary reasons of damages during earthquake in industries comes from the fire and electrical hazards from gas and electrical lines. An automatic load-disconnection system like this project could help in preventing such damages. Moreover, study of the earthquake data could help in identifying potentially vulnerable areas and take necessary measures.

5.5.3 Sustainability Evaluation (PO(g))

The project aims to develop a subscription-based system which lets users subscribe to our network and receive alerts on earthquake. Subscribers could also use our automatic load-disconnection feature, access earthquake data with date and time for further assessment. This model could be introduced to the industries under a subscription model as part of their safety measure. This could help our project be sustainable and profitable as well.

5.6 Ethical Issues (PO(h))

During our project, we communicated with Mr Saleh Hamza Priyo, research assistant at Department of Civil Engineering, BUET. Since he also developed a similar earthquake detection device, there were chances of collision in methodology. However, our discussions with Mr Saleh Hamza Priyo were limited to theoretical background of earthquake detection, testing methods etc. We also used a different sensor than him and integrated IoT platforms, load disconnections and other features which were not present in previous models. Our team consistently made sure that our project does not collide with any present models of the device and has space for novelty.

6 Reflection on Individual and Team work (PO(i))

6.1 Individual Contribution of Each Member

a) Tanvir Hossain (ID- 1906040) worked on the base code development for earthquake detection in ESP-32 along with ID-1906055. Moreover, he also implemented the noise filtering techniques like taking running average, use of capacitors etc. He also designed the casing for testing

b) Tasmin Khan (ID-1906055) worked on the sensor testing, integration and data accumulation for the correct choice of sensor for the project. She also worked on storing the accelerometer data locally into the SD card and on further modifications of the hardware circuit

c) Md Hasib Ur Rashid (ID-1906059) worked on the ThingSpeak IoT platform integration for data visualization along with ID-1906065. He also worked on load disconnection and development of an alert system for the earthquake.

d) A.K.M. Anindya Alam (ID-1906065) worked on the IoT communication between the two ESP-32 using Blynk IoT platform. He also worked on exporting the data from SD card to the IoT platform and development of the alert system along with ID-1906059.

6.2 Mode of TeamWork

The team was divided into two parts- hardware development and IoT integration. ID- 1906040 and ID-1906055 worked on the hardware development portion whereas ID-1906059 and ID-1906065 worked on IoT integration and data visualization. The team worked together on earthquake detection strategies and worked together throughout the span of the project for seamless integration of hardware and IoT. The team also sat together with the Civil Engineering Department of BUET on multiple occasions for constant development of the device.

6.3 Log Book of Project Implementation

Date	Milestone achieved	Individual Role	Team Role	Comments
17/12/2023	Meeting with RA of Civil Engineering Department	1906055 arranged meeting with RA after communication with Dr Zunaid Baten	The team sat together in an online meeting with RA and gained valuable insights about the project	
19/12/2023	Project Proposal		Literature study on earthquake detection for proposal	
30/12/2023	Basic hardware implementation		Team sat together for assembling the basic hardware	

			along with the sensors	
13/1/2023	Sensor testing	1906040, 1906055	Testing both of the sensors for choosing the more accurate one	
19/1/2023	IoT integration	1906065, 1906059	Integration of Blynk for successfully integrating the two microprocessors	
27/1/2023	Earthquake detection		First successful earthquake detection test	
7/2/2023	Storing data in SD Card	1906055, 1906040	Successfully storing the data locally in SD card	
9/2/2023	Second meeting with Civil Engineering Department		The team sat with the RA from Civil Engineering Department for further analysis on the device and possible use of shaker table.	
12/2/2023	Exporting SD card data from SD card module to IoT platform	1906065, 1906059		
21/2/2023	Integration of all IoT platforms and development of alert system	1906065	The other team members helped in developing the alert system in email and WhatsApp	
26/2/2023	Load Disconnection	1906040, 196059	The team sat together to implement load disconnection at a remote location using IoT	
29/2/2023	Casing design	1906040	The team worked on material selection and design strategies for the casing together	

7 Communication to External Stakeholders (PO(j))

7.1 Executive Summary

The early earthquake detection device uses accelerometer data from sensors to detect earthquake before the shockwave hits. After detection, it disconnects the loads connected to the system and sends an alert to the users of the model. The users can also access the data from an app made specially for the device and observe the accelerometer values, time of the earthquake and intensity of the earthquake as well. After 1.5 minutes of earthquake, it reconnects the load to the system again and stops the alert.

7.2 YouTube Link of Demonstration

The demonstration video has been uploaded in the link here.

7.3 Source Code Link

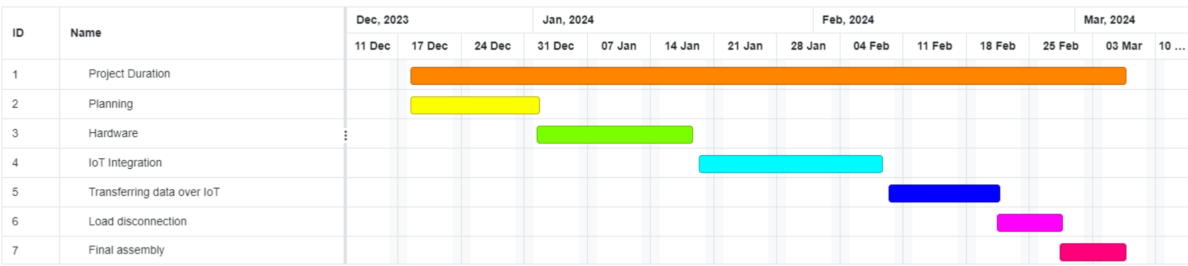
The Arduino code for this project has been uploaded in the link here

8 Project Management and Cost Analysis (PO(k))

8.1 Bill of Materials

Name of component	Price per unit (BDT)	Quantity	Total Price
Esp 32	675	2	1350
MPU6050	2250	1	2250
RTC Module	190	1	190
ADXL 335	890	1	890
SD Card Module	250	1	250
SD Card	400	1	400
Relay Module	290	1	290
Light bulbs and holders	250	2	500
Total			6120

8.2 Timeline of Project Implementation



9 Future Work (PO(l))

a) **Testing using Shaker Table:** A possible collaboration with Department of Civil Engineering, BUET could help in using the shaker table and analyzing the threshold for the device for further improvement

b) **Use of high-end sensors:** A more precise high-end sensor could help in more accurate detection of earthquakes

c) **Development of subscription model:** A subscription model for our system could help us expand the network and use high end IoT platforms and help the system be sustainable.

10 References

[1] https://www.tbsnews.net/bangladesh/bangladesh-records-6-earthquakes-exceeding-magnitude-5-january-749986

[2] Al Zaman MDA, Monira NJ (2017) A Study of Earthquakes in Bangladesh and the Data Analysis of the Earthquakes that were Generated in Bangladesh and its' Very Close Regions for the Last Forty Years (1976-2016). J Geol Geophys 6:300.

[3] Dorveaux, Eric & Vissiere, David & Martin, Alain & Petit, N.. (2010). Iterative calibration method for inertial and magnetic sensors. Proceedings of the 48th IEEE Conference on Decision and Control. 8296 - 8303. 10.1109/CDC.2009.5399503.

[4] Ahmed, Mohiuddin & Gopal, Chandra & Roy, & Das, Shishir & Hasan, Mehidi. (2019). Seismic Behavior of RCC Buildings under Revised Seismic Zone Classification by BNBC. 9. 21-28.

[5] S. Kim, I. Khan, S. Choi and Y. -W. Kwon, "Earthquake Alert Device Using a Low-Cost Accelerometer and its Services," in *IEEE Access*, vol. 9, pp. 121964-121974, 2021, doi: 10.1109/ACCESS.2021.3103505.