Punjab Engineering College Chandigarh

Major Project Report

Wireless Ground Vibration Sensor System Design

Major Project

Wireless Ground Vibration Sensor System Design

Submitted by

Omkar Singh Sandhu	**20105004**
Akshat Kumar	**20105005**
Agrim Gupta	**20105008**
Rahul Sharma	**20105015**
Kanwar Ekamjot Singh	**20105021**

Under the Guidance of

Dr. Jyoti Kedia

Declaration

We, the undersigned, solemnly declare that the project report titled "Wireless Ground Vibration Sensor System Design" represents an authentic and original account of our work undertaken during the 7th Semester of our Bachelor's degree in Electronics and Communication Engineering at Punjab Engineering College. This project was conducted under the esteemed guidance of Dr. Jyoti Kedia. We assert that the contents of this report are the result of our independent investigation, and we have not submitted this work for any other academic purpose. We acknowledge all sources of information and data used in the report through proper citation. The project has been executed in accordance with the stipulated guidelines of the college. We affirm that our work does not violate any existing copyright, patent, or proprietary rights. Contributions from external sources have been duly acknowledged. We recognize the consequences of plagiarism and assert that our work is entirely original. This declaration underscores our commitment to the integrity and authenticity of the project work.

Omkar Singh Sandhu	**20105004**
Akshat Kumar	**20105005**
Agrim Gupta	**20105008**
Rahul Sharma	**20105015**
Kanwar Ekamjot Singh	**20105021**

Date: 13/05/2024

Certified that the above statement made by the students is correct to the best of our knowledge and belief.

Dr. Jyoti Kedia

Department of Electronics and Communication Engineering

Punjab Engineering College

Table of Contents

Acknowledgement

Acknowledgement

We extend our deepest gratitude to Prof. Jyoti Kedia for their unwavering support, guidance, and indispensable assistance, without which this project would not have come to fruition. Prof. Kedia's insightful feedback and direction played a pivotal role in shaping the overall quality and success of this report. Their mentorship has been invaluable, and we are truly thankful for their contribution.

Our sincere thanks go to Prof. Arun Kumar Singh for generously providing us the time and resources to work in the department labs. This opportunity allowed us to fully immerse ourselves in our work and contributed significantly to the project's success.

We would also like to express our acknowledgment to our fellow group members. The dedication, commitment, and collaborative spirit each member brought to the table were instrumental in the completion of this project. The synergy and effective teamwork exhibited by the group truly made a difference in the final outcome, and we are grateful for their contributions.

Finally, we acknowledge the Electronics and Communication Department at Punjab Engineering College. Their provision of necessary resources, facilities, and funding played a crucial role in conducting this project. Without their unwavering support, this endeavour would not have been possible.

To everyone who has been a part of this journey, thank you for your invaluable support. Your contributions have been integral to the success of this project, and we are grateful for the opportunity to have worked alongside such dedicated individuals.

Omkar Singh Sandhu	**20105004**
Akshat Kumar	**20105005**
Agrim Gupta	**20105008**
Rahul Sharma	**20105015**
Kanwar Ekamjot Singh	**20105021**

Abstract

Abstract

The project presents an innovative solution for ground vibration monitoring during blasting operations through the development of a wireless sensor system. Traditionally, wired seismic sensors have been employed, but their limitations in terms of setup complexity, maintenance demands, and scalability have prompted the exploration of alternative technologies. Building upon the findings of the University of Kentucky's thesis report, our project focuses on overcoming the drawbacks associated with traditional seismographs.

The core components of our wireless sensor system include the SM-24 Geophone Sensor, known for its precision and low-distortion characteristics, and the ADS1115 16-bit Analog-to-Digital Converter for high-resolution data conversion. An ESP32 microcontroller manages data acquisition and transmission, while EspNow protocol provides wireless communication. The system is integrated with a Esp32, serving as a data hub and facilitating the creation of a user-friendly graphical user interface (GUI).

The methodology involves precise ground vibration measurement using the geophone sensor, analog-to-digital conversion, and wireless transmission of data to a central hub. The project aims to address the challenges of wired sensors by introducing wireless capabilities, improving accuracy through geophone sensors, and enabling real-time data retrieval and analysis.

The proposed system aims to revolutionize ground vibration monitoring by offering a cost-effective, scalable, and efficient alternative to traditional wired sensors. The integration of advanced technologies not only enhances accuracy in ground vibration detection but also facilitates seamless data access, fostering collaboration and informed decision-making. The project's outcomes are expected to contribute significantly to the fields of blasting research and beyond, with potential applications in various industrial and research domains.

Introduction

Introduction: Wireless Ground Vibration Sensor Project

In contemporary industrial landscapes, the dynamic interplay between human activities and the Earth's crust often gives rise to ground vibrations, a phenomenon that demands meticulous monitoring and analysis. Ground vibrations, generated by activities such as blasting, construction, and mining, have far-reaching implications on structural integrity, environmental sustainability, and human safety. As industries strive for efficiency, scalability, and cost-effectiveness, there arises a critical need to innovate in the realm of ground vibration sensing technologies.

This project embarks on a pioneering endeavor to revolutionize ground vibration monitoring through the development of a Wireless Ground Vibration Sensor System. Traditional approaches to ground vibration sensing have predominantly relied on wired seismic sensors and seismographs, entailing cumbersome setups, high maintenance costs, and limited scalability. The Wireless Ground Vibration Sensor Project seeks to transcend these limitations, introducing a paradigm shift in the way ground vibrations are detected, analyzed, and communicated.

Understanding Ground Vibrations:

Ground vibrations encompass a spectrum of oscillations resulting from both continuous and transient forces acting on the Earth's surface. Whether induced by the rhythmic operation of heavy machinery or sudden explosive events like blasting, these vibrations manifest as mechanical waves propagating through the ground. The project recognizes the multifaceted nature of ground vibrations, including low-frequency oscillations inherent in construction activities and high-frequency vibrations characteristic of explosive events or seismic disturbances.

Types of Ground Vibrations:

Continuous Vibrations: Characterized by steady-state oscillations resulting from prolonged industrial activities, such as machinery operation or continuous industrial processes.

Transient Vibrations: Abrupt, short-lived oscillations typically associated with impulsive events like explosions, seismic disturbances, or rapid ground changes.

Low-Frequency Vibrations: Vibrations occurring at frequencies below 2 Hz, often linked to construction, transportation, or industrial operations with long-lasting impacts.

High-Frequency Vibrations: Vibrations occurring at frequencies above 2 Hz, prevalent in dynamic events like blasting operations, earthquakes, or other instantaneous disturbances.

Surface Waves: Travel along the Earth's surface and include Love waves and Rayleigh waves, causing horizontal and elliptical ground motion, respectively.

Body Waves: Travel through the Earth's interior and include Primary (P) and Secondary (S) waves, with P waves causing compressional ground motion and S waves causing shearing motion.

Traditional Sensing Methods:

Historically, the conventional approach to ground vibration monitoring involved the use of wired seismic sensors and seismographs. Seismographs, designed to record and measure ground motion, incorporate geophones or accelerometers as ground motion sensors. Geophones, known for their precision, and accelerometers, measuring proper acceleration, have been integral to capturing ground vibrations accurately. While these methods have been effective, their reliance on extensive wiring, susceptibility to environmental wear and tear, and limited scalability have prompted a quest for innovative alternatives.

The Wireless Ground Vibration Sensor Project:

This project signifies a departure from traditional wired sensing methods, envisioning a Wireless Ground Vibration Sensor System that eliminates the constraints associated with physical connectivity. Leveraging cutting-edge technologies, including geophone sensors, analog-to-digital conversion, and 2.4Ghz wireless communication, the wireless system promises enhanced accuracy, scalability, and real-time data access.

Applications of the Wireless Ground Vibration Sensor:

The Wireless Ground Vibration Sensor system holds the potential to transform various industries and applications:

- Construction and Mining: Ensure safety and structural integrity during construction and mining operations.

- Blasting and Demolition: Assess and mitigate the impact of explosive events on surrounding structures.

- Environmental Monitoring: Evaluate the environmental impact of industrial activities on ecosystems.

- Seismic Research: Contribute valuable data to seismological research for earthquake and seismic event studies.

Conclusion of the Introduction:

The Wireless Ground Vibration Sensor Project signifies a leap forward in ground vibration monitoring technology. By addressing the limitations of traditional wired systems and harnessing the power of wireless communication and advanced sensor technologies, the project aims to provide an efficient, scalable, and cost-effective solution. As the project unfolds, it holds the promise of not only advancing the field of ground vibration monitoring but also ushering in a new era of innovation with far-reaching implications for diverse industries and applications worldwide.

Problem Statement

Problem Statement

The existing paradigm for detecting ground vibrations during blasting operations relies predominantly on the utilization of wired seismic sensors. Although these sensors play a pivotal role in ensuring safety and compliance, they are burdened by significant limitations that impede overall efficiency, scalability, and cost-effectiveness in the field. This problem statement endeavors to comprehensively articulate the critical challenges associated with the current wired seismic sensor technology.

1. Cumbersome Setup:

Traditional wired seismic sensors necessitate a labor-intensive and time-consuming setup process. This includes the meticulous deployment of sensors at precise locations, the intricate laying of extensive wiring networks, and the installation of power sources. This intricate setup consumes valuable resources, both in terms of time and manpower, imposing a substantial logistical burden on operational workflows.

2. High Maintenance:

Wired sensor systems are inherently susceptible to wear and tear owing to their reliance on physical wiring components. The perpetual exposure to harsh environmental conditions, ranging from extreme temperatures to moisture, engenders frequent maintenance requirements. This recurrent need for upkeep results in heightened operational downtime and increased costs, adversely impacting the overall effectiveness of ground vibration monitoring systems.

3. Limited Scalability:

Expanding a wired sensor network to cover larger areas or to monitor additional locations poses a logistical challenge. Each new sensor installation necessitates the extension of wires, a process that is labor-intensive and economically burdensome. The limitations in scalability impede the adaptability of the system to dynamic operational requirements, hindering its efficacy in diverse settings.

4. Inefficient Data Retrieval:

Data retrieval from wired sensors typically demands physical access to each sensor point, rendering the process less efficient for real-time monitoring and data analysis. This inherent limitation obstructs timely decision-making, particularly in applications where immediate action is imperative. The dependence on physical access points introduces delays in the interpretation and utilization of collected data, diminishing the system's responsiveness.

5. Increased Operational Costs:

The amalgamation of intricate setup procedures, frequent maintenance demands, and limited scalability contributes to elevated operational costs over time. These cumulative costs strain budgets and resources, potentially jeopardizing the feasibility and sustainability of ground vibration monitoring projects. The financial implications associated with the current wired sensor technology underscore the urgent necessity for a more cost-effective and sustainable alternative.

To surmount these formidable challenges, there exists a compelling need for an innovative solution that can provide wireless, low-maintenance, scalable, and cost-effective ground vibration detection technology. This project endeavors to address this imperative by developing a modern wireless seismic sensor system, poised to transcend the limitations of traditional wired sensors. The envisaged solution aims to ensure not only the accuracy and reliability of data collection but also the seamless integration of advanced monitoring capabilities for blast operations and other pertinent applications.

Objective

Objective

The primary aim of this project is to build upon the findings and limitations identified in the University of Kentucky's thesis report titled "Proof of Concept for the Development of a Ground Vibration Sensor System for Future Research in Blasting." This comprehensive exploration investigated alternative methods of ground vibration detection, with a particular focus on piezoelectric sensors. However, the research revealed drawbacks in signal quality, signal-to-noise ratio, and data correlation when compared to traditional seismographs, especially in the context of high-frequency ground vibrations.

In contrast to the University of Kentucky's approach, our project seeks to address the inherent limitations of traditional seismographs rather than replacing the sensor technology for vibration detection. While piezoelectric sensors offer cost advantages and suitability for large-scale deployments, they come with limitations such as variable accuracy, non-uniform signal quality, and sensitivity to signal angles. In contrast, triaxial geophone sensors, akin to those found in professional seismographs, provide superior data quality and directional information.

The key objectives of our project can be summarized as follows:

1. Wireless Sensor System:

Develop a state-of-the-art wireless ground vibration sensor system that eliminates the need for extensive cabling and physical access to individual sensors. This enhancement aims to streamline deployment processes, increase scalability, and reduce logistical complexities.

2. Geophone Sensor Integration:

Utilize advanced geophone sensors to maintain high levels of accuracy in ground vibration detection. Drawing inspiration from professional seismographs, this integration is designed to overcome the limitations associated with alternative sensor technologies, providing a robust and reliable solution.

3. Improved Data Retrieval:

Implement innovative data retrieval techniques to enable real-time monitoring and analysis. This objective focuses on ensuring more timely decision-making in applications that demand immediate attention, overcoming the historical constraint of physical access to individual sensors for data extraction.

4. Cost Efficiency:

Develop a cost-effective solution that significantly reduces operational expenses associated with ground vibration monitoring. By enhancing efficiency and scalability while leveraging advanced sensor technologies, this objective aims to make ground vibration monitoring more accessible and sustainable for various research and industrial applications.

Through the pursuit of these objectives, our project aspires to revolutionize ground vibration monitoring by offering a wireless, accurate, and cost-effective alternative to traditional seismographs. This innovative solution is anticipated to have wide-ranging applications in research related to blasting and beyond, promising significant advancements in data collection, analysis, and decision-making processes. The project's success is poised to contribute to transformative developments in the field of ground vibration monitoring, paving the way for more effective and sustainable practices in various industrial and research domains.

Research

Proof of Concept for the Development of a Ground Vibration Sensor System for Future Research in Blasting

Inferences from the Research Paper:

1. Need for Upgraded Technology:

- The mining industry utilizes traditional and outdated devices for vibration monitoring, leading to a significant technological lag.
- The absence of specific guidelines for equipment requirements and post-processing analysis hampers accurate research in ground vibrations.

2. Commercial Device Limitations:

- Existing commercial devices are both expensive and bulky, suitable mainly for long-term monitoring at fixed locations.
- These devices are not adaptable to large-scale research projects required to comprehend vibration transmission and structural responses fully.

3. Opportunities in Technological Advancements:

- Rapid growth in electronics and piezoelectric materials presents an opportunity to update systems and sensors used in the mining industry.
- New technologies can potentially lead to a reevaluation of vibration regulations in the mining sector.

4. Purpose of the Study:

- The research aims to outline a developed vibration system for monitoring blast vibrations, offering an alternative to currently available systems.
- Two primary research questions are posed:
  - Can the assembled system effectively measure and record blast vibrations from a surface mining operation?
  - Is the acquired data accurate and precise enough for future research projects in mining?

5. Procedures and Methodology:

- A prototype system was developed, configured to replicate sampling rates used in the mining industry.

- Ground vibrations from surface blasts in a coal mine were studied using the prototype system, a commercial seismograph, and an accelerometer.

- The study employed cross-correlation, visual comparison, and spectral coherence for waveform analysis.

6. Limitations of the Study:

- The study used components from electronic manufacturers, and their configurations were confined to the original purposes of those devices.

- Lack of control over blast types and locations due to ongoing mining operations resulted in minimal lead time for data acquisition.

- The developed system represents the first iteration, emphasizing the need for further testing and the feasibility of a more comprehensive monitoring system.

7. Organization of the Study:

- The thesis is structured to provide background information on ground vibration response, types of sensors, and their role in studying ground vibrations.

- The developed prototype system and the University of Kentucky's vibration project are outlined, followed by data analysis, concluding remarks, and recommendations for future work.

Methodology

Methodology

This section elucidates the intricate methodology underpinning our project, centering on the development of a wireless ground vibration sensor system utilizing geophone sensors, analog-to-digital conversion, and 2.4Ghz wireless communication technology. The following steps meticulously delineate our innovative approach:

1. Geophone Sensor for Ground Vibration Measurement:

Our project employs a geophone sensor as the primary device for ground vibration measurement. Geophones are selected for their recognized precision and reliability in detecting ground vibrations with utmost accuracy.

2. Analog-to-Digital Conversion (ADC):

The analog data generated by the geophone sensor undergoes transmission to a 16-bit analog-to-digital converter (ADC). This pivotal component plays a crucial role in converting the analog signals into 16-bit digital data, ensuring high-resolution measurements that capture nuanced variations in ground vibrations.

3. ESP32 Microcontroller Integration:

Subsequently, the digital data from the ADC is relayed to an ESP32 microcontroller board. The ESP32 serves as the central processing unit for data acquisition and transmission within our system, facilitating efficient processing and management of the acquired data.

4. Data Transmission Using 2.4Ghz wireless Technology:

To facilitate wireless data transmission, our system incorporates 2.4Ghz wireless communication modules. The ESP32, equipped with a EspNow, transmits the digital vibration data over expansive distances to a designated receiving station.

5. Receiving and Data Processing:

At the receiving end, another 2.4Ghz module, integrated with a Esp32, captures the transmitted data packets. The Esp32 assumes the role of a data hub and executes the following tasks:

- Data Reception: It systematically collects incoming data packets from the 2.4Ghz module.

- Data Computation: The received data undergoes meticulous processing and analysis for various parameters, including amplitude, frequency, and duration of ground vibrations.

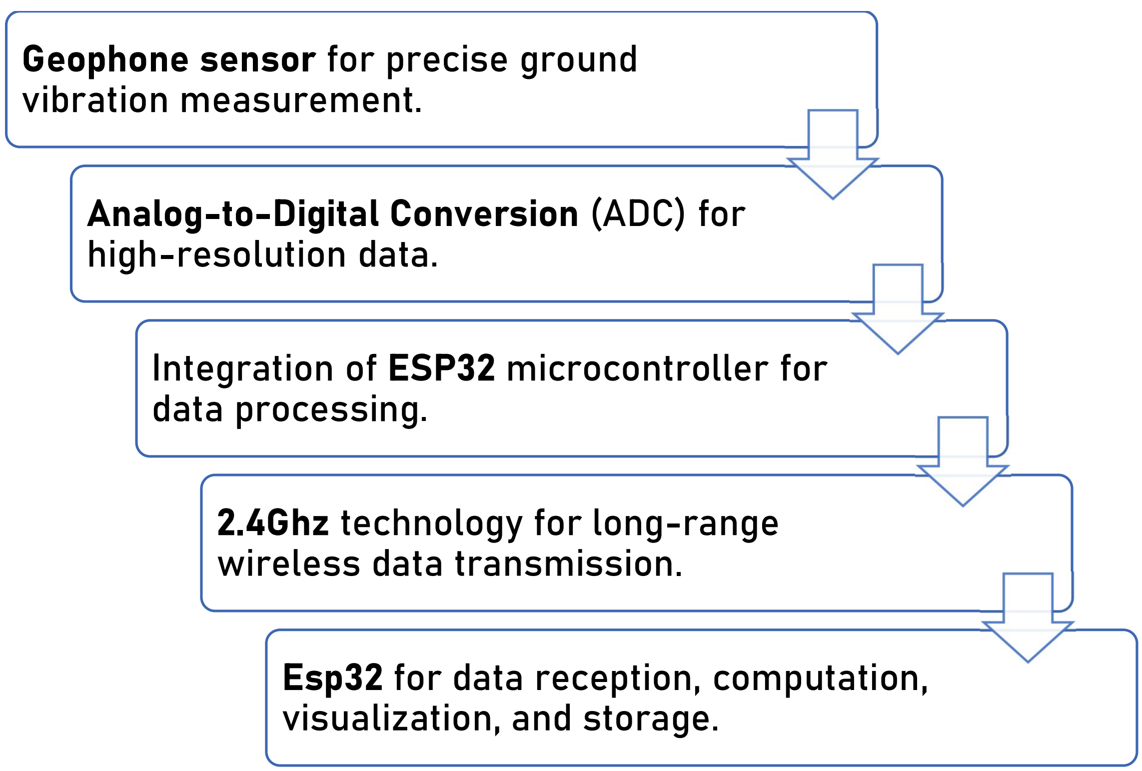
- Seismic Graph Generation: Real-time seismic graphs and visual representations of ground vibrations are generated for immediate visualization and comprehensive analysis.

- Data Storage: Processed data is stored for subsequent reference and in-depth analysis.

6. Wireless Data Access:

The stored data is made conveniently accessible wirelessly within the organization. Authorized personnel can seamlessly retrieve, view, and analyze ground vibration data from multiple systems, fostering collaborative research and facilitating informed decision-making.

This methodology synergistically combines the precision of geophone sensors, the high-resolution capabilities of a 16-bit ADC, and the long-range wireless communication attributes of 2.4Ghz technology. The resulting system not only ensures the accurate measurement and analysis of ground vibrations but also enables facile and efficient access to the collected data for various organizational needs. This comprehensive approach enhances research capabilities and facilitates informed decision-making, thereby positioning our project at the forefront of ground vibration monitoring technology.



Tools Used

1. Geophone Sensor – SM-24 Geophone Sensor

- Overview: The SM-24 Geophone Sensor stands out as a tight specification, low-distortion geophone designed for seismic exploration. It boasts an extended spurious range over 240 Hz, enabling full bandwidth at 2-ms sampling, making it a preferred choice for high-fidelity data collection in 2-D and 3-D surveys.

- Advantages:

  - Precise Measurement: The geophone's tight specifications and unique element design contribute to its precision in detecting ground vibrations.

  - Low Lifecycle Cost: The exceptional quality of the SM-24 makes it the geophone with the lowest lifecycle cost of ownership in the industry.

- Application: This geophone is pivotal in ensuring accurate and reliable ground vibration measurements, overcoming the limitations of alternative sensor technologies.

2. Analog to Digital Converter – ADS1115 16-bit ADC

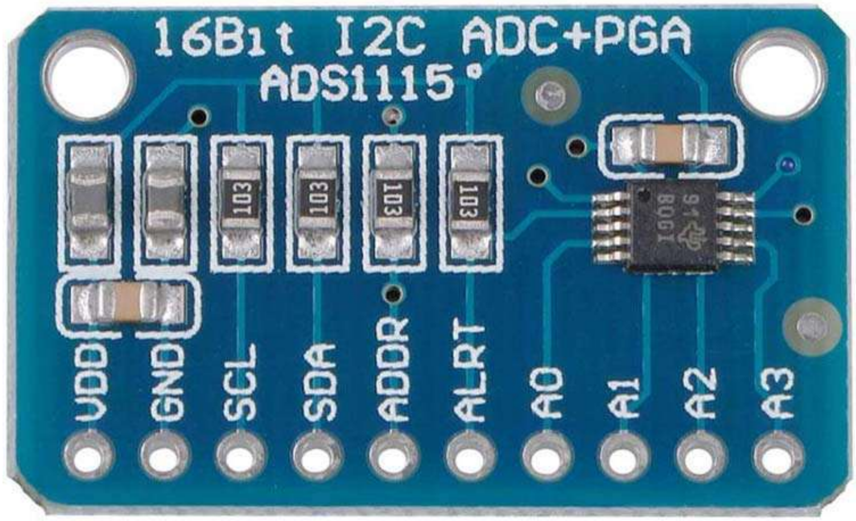
- Overview: The ADS1115 is a 16-bit analog-to-digital converter equipped with four analog channels. It facilitates the conversion of analog signals into high-resolution 16-bit digital data, offering 32,768 possible output values for every analog input.

- Advantages:

  - High Resolution: The 16-bit resolution provides a high level of detail in capturing variations in voltage.

  - Programmable Gain Amplifier: The PGA allows customization of reference voltage, enhancing accuracy.

- Application: The ADS1115 is integral in transforming analog signals from the geophone into digital data, ensuring precise and detailed measurements for further processing.



3. ESP32 Microcontroller – ESP32 Espressif Board

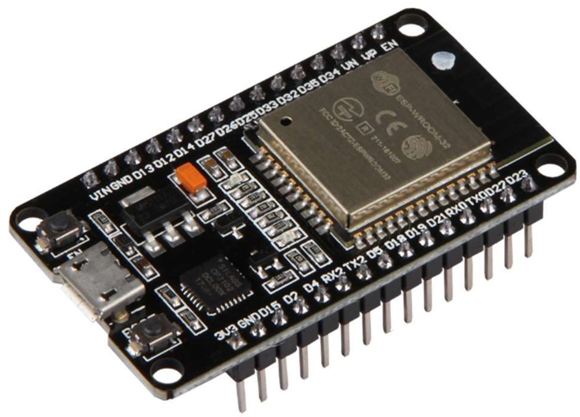
- Overview: The ESP32 microcontroller, part of the Espressif Board family, is characterized by its low cost, low power consumption, and rich peripheral input/output interface. It has 448 KB ROM for booting and core functions, and 520 KB SRAM for data and instructions.

- Advantages:

  - Low-Cost and Low-Power: The ESP32 is cost-effective and energy-efficient, making it suitable for diverse applications.

  - Rich Peripheral Interface: Its extensive support for various input and output peripherals enhances versatility.

- Application: Serving as the central processing unit, the ESP32 manages data acquisition, processing, and transmission within the wireless ground vibration sensor system.



3. 18650 Battery – 18650 Lithium ion Samsung

- Overview: The 18650 battery, a commonly used lithium-ion rechargeable cell, is renowned for its compact size, high energy density, and reliability. Named after its dimensions, 18mm by 65mm, this cylindrical cell has become a standard power source in various electronic devices due to its robust performance and longevity.

- Advantages:

    - High Energy Density: The 18650 battery packs a considerable amount of energy into its compact form factor, making it ideal for applications requiring prolonged usage without frequent recharging.

    - Longevity and Reliability: With a robust design and consistent performance over multiple charge cycles, the 18650 battery offers a dependable power solution for diverse electronic devices.

- Application: Serving as a primary or backup power source, the 18650 battery is employed in a myriad of applications, including portable electronics, electric vehicles, and renewable energy systems, providing sustained power for extended periods.

4. Software Tools Used for Webpage Design:

- Python Programming: Python is employed for sensor data collection and GUI development due to its versatility and ease of learning.

- ReactJS: ReactJS, an open-source JavaScript library, is utilized for building dynamic and responsive user interfaces in the GUI.

- MongoDB: MongoDB, a NoSQL database system, is chosen for its flexibility and scalability in handling unstructured and semi-structured data, supporting efficient data storage and retrieval.

In summary, the amalgamation of these tools ensures a robust, cost-effective, and efficient wireless ground vibration sensor system. From the precise measurement capabilities of the geophone sensor to the energy-efficient long-range communication facilitated by 2.4Ghz modules, each tool plays a crucial role in achieving the project's objectives. The ESP32 microcontroller orchestrates the data flow, and the Esp32 serves as the central processing unit and GUI platform, creating a seamless and comprehensive solution for ground vibration monitoring.

Tentative Budget

Tentative Budget

Component	Specifications	Tentative Cost
Geophone Sensor	**SM-24 Geophone Sensor**	**7600**
Analog to Digital Converter	**ADS1115 16-bit ADC**	**700**
Microcontroller	**Esp32 Espressif Board**	**1200**
18650 battery	**18650 Lithium ion battery Samsung**	**160**
PCB	**80mm*80mm, 2 layer PCB**	**1650**
3D Printed Case	**Custom Designed Case**	**400**
Total		**11710**

Work Done

Topic Research

Geophone

SM-24 Geophone Element

Where Quality Data Starts

Features

- Tight specification, low-distortion geophone
- Extended spurious over 240 Hz, allowing full bandwidth at 2-ms sampling
- Backwards compatible with SM-4, SM-4 SuperphoneTM range, and SM-24ST
- Horizontal element available for shear-wave and 3-C recording.
- 3-year non-prorated warranty
- Lowest lifecycle cost of ownership in the industry
- Installed base of over 8 million worldwide (est.)

The SM-24 geophone element is designed to offer the highest performance in seismic exploration based upon field-proven I/O Sensor technology. Low distortion, combined with excellent specifications, provide high-fidelity data in 2-D and 3-D surveys. The extended bandwidth allows the full potential of 2-ms/24-bit recording systems to be realized.
The tight specifications, unique element design, and exceptional quality of the Sensor SM-24 make it the lowest life cycle cost of ownership geophone in the industry.

Applications: 2-D & 3-D seismic exploration with bandwidth from 10 Hz up to 240 Hz.

Implementation: Can be installed in a variety of I/O Sensor geophone cases.

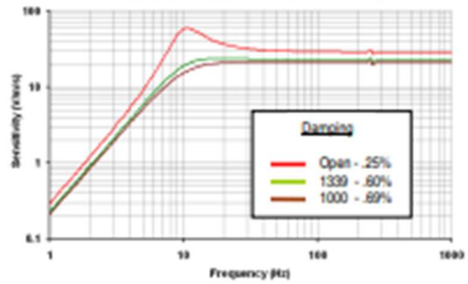
I/O

SENSOR Nederland b.v.

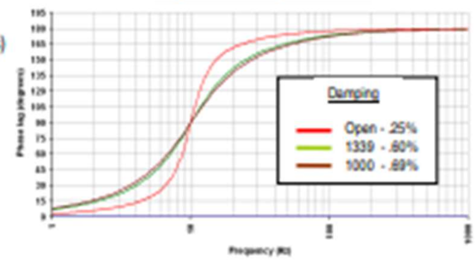
Specifications: SM-24 Geophone Element

Frequency	
Natural frequency	10 Hz
Tolerance	± 2.5%
Maximium tilt angle for specified Fn	10°
Typical spurious frequency	>240 Hz
Distortion	
Distortion coil to case velocity with 17.78mm/s (0.7 in/s) p.p.	<0.1%
Distortion measurement frequency	12 Hz
Typical distortion (string of 12 in series, measured at 12 Hz)	0.03 %
Damping	
Open circuit (typical)	0.25
Damping calibratio- shunt resistance	1,339 Ω
Damping with calibration shunt	0.6
Tolerance with calibration shunt	+ 5 %, - 0 %
Sensitivity	
Sensitivity	28.8 V/m/s (0.73 V/in/s)
Tolerance	± 2.5 %
RtBcfn	6,000 Ω Hz
Moving mass	11 g (0.38 oz)
Max coil excurstion p.p.	2 mm (0.08 in)
Coil Resistance	
Standard	375 Ω
Tolerance	+/- 2.5 %
Physical Characteristics	
Diameter	25.4 mm (1 in)
Height	32 mm (1.26 in)
Weight	74 g (2.6 oz)
Operating temperature range	-40°C to +100°C (-40°F to +212°F)
Warranty Period*	3 years

Geophone Response Curve – SM-24 10 Hz



Geophone Phase Lag – SM-24 10 Hz



* Warranty excludes damage caused by high voltage and physical damage to the element case.

All parameters are specified at +20°C in the vertical position unless otherwise stated.

Ordering Information

SM-24

SM-24/U-B 10 Hz 375 Ohm
(upright basic unit with insulating disc) P/N 1004117

Esp32 and ADC

Why ESP32:

- Low-cost: Accessible starting at $6, making it affordable for a wide audience.

- Low-power: Consumes minimal power, supports low-power modes like deep sleep.

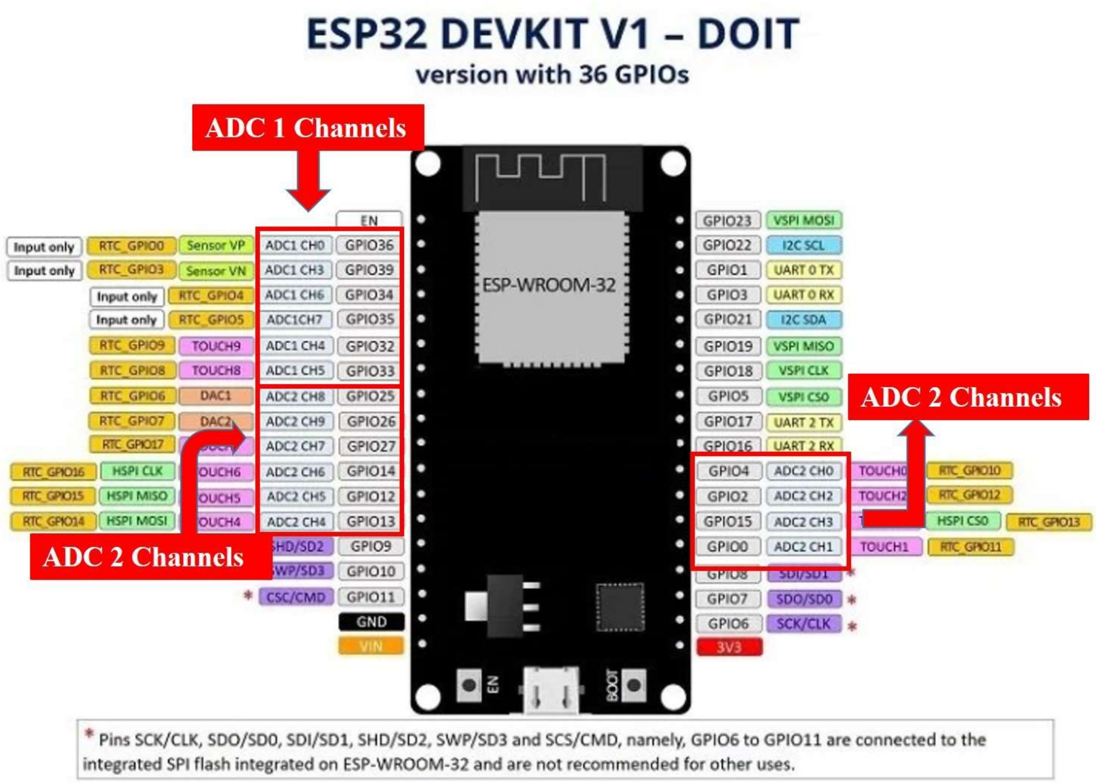
- Wi-Fi capabilities: Easily connects to Wi-Fi networks, crucial for IoT and Home Automation projects.

- Bluetooth: Supports both Bluetooth classic and Bluetooth Low Energy (BLE) for diverse IoT applications.

- Rich peripheral I/O interface: Offers a variety of input/output peripherals such as touch, ADCs, DACs, UART, SPI, I2C, PWM, etc.

- Arduino compatibility: Can be programmed using the Arduino programming language.

- MicroPython compatibility: Allows programming with MicroPython firmware, catering to Python enthusiasts.



ESP32 Specifications:

- Wireless connectivity: Wi-Fi with a data rate of 150.0 Mbps and BLE support.

- Processor: Tensilica Xtensa Dual-Core 32-bit LX6 microprocessor running at 160 or 240 MHz.

- Memory: ROM, SRAM, RTC fast SRAM, RTC slow SRAM, eFuse, and embedded flash options.

- Low Power: Supports ADC conversions even during deep sleep.

- Peripheral Input/Output: Diverse peripherals like capacitive touch, ADCs, DACs, UART, SPI, I2C, PWM, etc.

- Security: Hardware accelerators for AES and SSL/TLS.

Why External ADC (ADS1115):

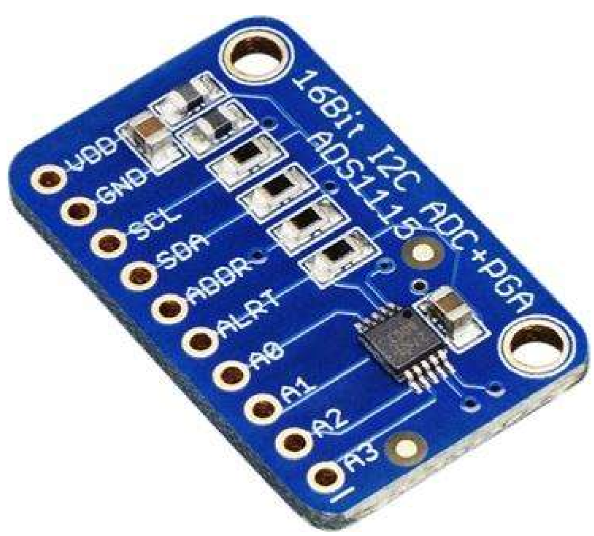
- Channel limitations: ESP32's built-in ADC modules may have channel limitations due to overlapping functionality.

- Non-linear characteristics: ESP32's internal ADC may exhibit non-linear behavior during voltage measurements.

- Low resolution: ESP32's ADC may have low resolution, making it challenging for accurate results.

- Resolution improvement: External ADC, such as ADS1115, offers higher resolution (16-bit) and accuracy.

- Versatility: ADS1115 can measure both positive and negative voltages, providing flexibility.



ADS1115 Features:

- 16-bit I2C ADC: Four analog channels, providing accurate measurements.

- Output format: Provides output in signed integer format.

- Resolution: Offers a resolution of 0.1875mV, enhancing accuracy compared to ESP32's built-in ADC.

- Addressing modes: Can be interfaced with different microcontrollers using various addressing modes.

- Library support: Adafruit provides a library for easy integration with Arduino IDE.

Pinouts of ADS1115

Pin name	Functionality
Vdd (Power supply pin)	Connect power supply between 2.2 – 5.5 volts
GND (Common reference pin)	Connect with ground pin of power supply
SCL	I2C SCL (serial clock pin)
SDA	I2C SDA (serial data pin)

ADDR	I2C slave select pin or address
ALRT	Alert/Ready
A0	Analog channel 0
A1	Analog channel 1
A2	Analog channel 2
A3	Analog channel 3

Interfacing ESP32 and ADS1115:

1. Connections:

- Connect VDD to VDD, GND to GND, SDA to GPIO21, SCL to GPIO22, ADDR to GND, and A0 to the analog voltage signal.

- Utilize GPIO pins 21 and 22 for I2C communication.

2. Software:

- Use Arduino IDE to program ESP32.

- Install the Adafruit ADS1115 library for I2C communication.

- Write code to handle communication and receive measured voltage over I2C protocol.

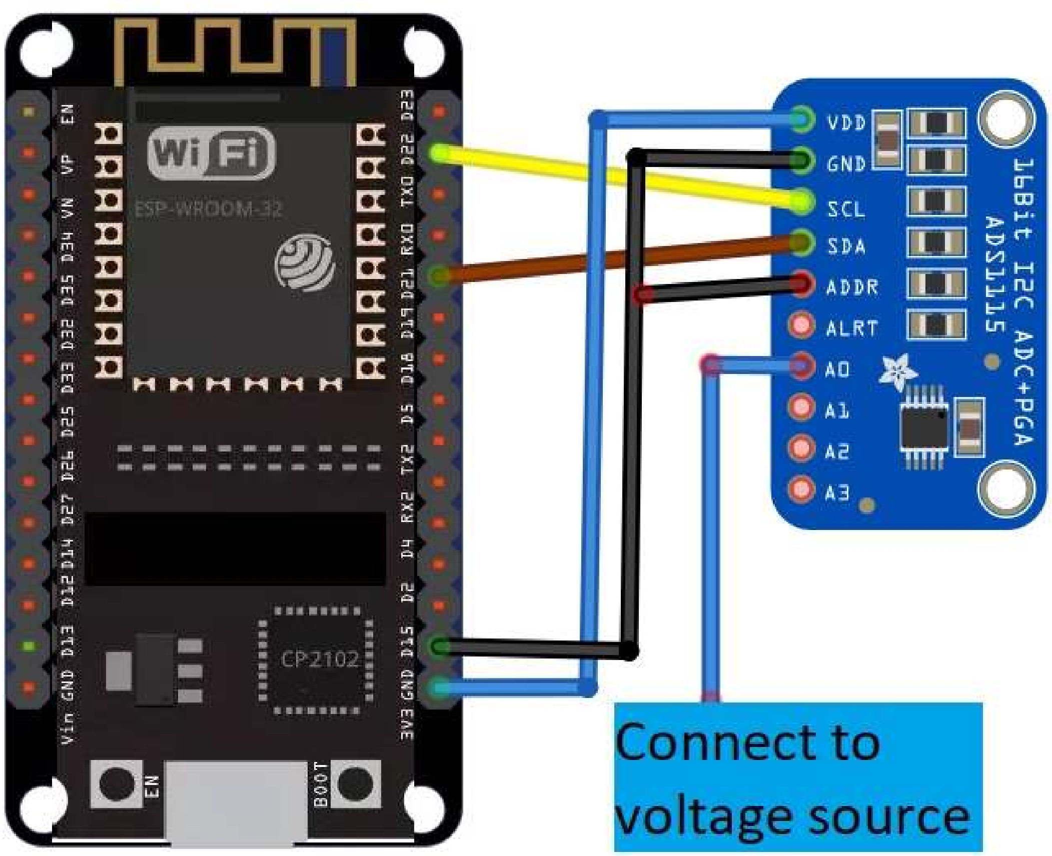
3. Purpose:

- Measure analog voltages with high accuracy using ADS1115.

- Overcome limitations of ESP32's built-in ADC.

- Enhance resolution and accuracy in voltage measurements.

- Enable diverse applications in IoT and sensor-based projects.



ESP32	ADS1115 external ADC
VDD	VDD
GND	GND
GPIO21 (SDA Pin)	SDA
GPIO22 (SCL Pin)	SCL
GND	ADDR
Analog voltage signal	A0

ESPNow Transmission

Why ESP Now for Data Transmission:

- Low latency communication: ESPNow offers low-latency communication suitable for real-time applications such as IoT sensor networks and home automation.

- High throughput: Provides high data transfer rates, enabling efficient transmission of large datasets or multimedia content.

- Low power consumption: ESPNow protocol is optimized for low power consumption, making it suitable for battery-operated devices and energy-efficient systems.

- Direct device-to-device communication: Enables direct communication between ESP32 devices without the need for a traditional Wi-Fi network, simplifying network architecture and reducing overhead.

- Mesh networking capabilities: ESPNow supports mesh networking, allowing devices to relay data through intermediate nodes, extending the communication range and improving reliability.

- Robustness: Offers robust communication even in noisy RF environments, ensuring reliable data transmission in various deployment scenarios.

ESP32 Module and ESPNow Connection

- ESP-IDF Integration: ESPNow is integrated into the ESP32 IDF (IoT Development Framework), providing native support for ESP32 devices.

- Simple Setup: ESPNow communication between ESP32 devices can be established with minimal configuration, requiring only the MAC address of the receiver and a shared encryption key.

- Peer-to-peer Communication: ESPNow facilitates direct peer-to-peer communication between ESP32 devices, enabling efficient data exchange without the need for a centralized server.

- Asynchronous Communication: ESPNow operates asynchronously, allowing devices to transmit and receive data independently, maximizing throughput and minimizing latency.

- Scalability: ESPNow communication can scale to accommodate large networks of ESP32 devices, making it suitable for applications ranging from simple sensor networks to complex IoT deployments.

Required Libraries for Arduino IDE:

1. ESP8266 and ESP32 Oled Driver for SSD1306 display: Library for interfacing with SSD1306 OLED displays on ESP8266 and ESP32 platforms.

   - Library Link: https://github.com/ThingPulse/esp8266-oled-ssd1306

2. ESP8266 and ESP32 Oled Driver for Adafruit-GFX: Graphics library used in conjunction with the SSD1306 OLED display driver for ESP8266 and ESP32.

   - Library Link: https://github.com/ThingPulse/esp8266-oled-ssd1306

3. One Wire Library: Library for working with one-wire communication protocol.

   - Library Link: https://github.com/PaulStoffregen/OneWire

4. DallasTemperature Library: Library for interfacing with Dallas temperature sensors.

   - Library Link: https://github.com/milesburton/Arduino-Temperature-Control-Library

Connections:

- ESP32 to OLED Display: Connect the ESP32 to the SSD1306 OLED display using the appropriate pins for communication (I2C or SPI), power, and ground connections.

- Library Integration: Include the mentioned libraries in the Arduino IDE for successful compilation of the code.

Research Papers study:

1. A Study of ESPNow Low Power and Local Area Network Technology

Inferences - The paper examines the performance of ESPNow LPN technology, emphasizing its low-power characteristics and suitability for local area network applications. It explores various parameters impacting data transmission and network efficiency in ESPNow-based systems.

2. ESPNow Transmission Parameter Selection

Inferences - The paper investigates the influence of ESPNow transmission parameters on communication effectiveness in IoT deployments. It proposes strategies for optimizing parameter selection to achieve a balance between network performance and power consumption in ESPNow networks.

3. Enhancing the Performance of ESPNow through Adaptive Data Rate Adjustment

Inferences - The paper explores methods for improving the performance of ESPNow by dynamically adjusting data rates based on environmental conditions and network congestion. It highlights the potential for enhancing range and reliability in ESPNow-based systems through adaptive data rate control.

Parameters for LoRa Transmission

To optimize the performance of ESPNow communication, we can adjust various parameters to achieve the desired balance between range, data rate, and reliability.

1. Channel:

- ESPNow operates on Wi-Fi channels, allowing for flexibility in channel selection to avoid interference and optimize communication range and stability.

2. Data Rate:

- ESPNow supports configurable data rates, allowing for adjustments to suit specific application requirements. Higher data rates offer increased throughput but may result in reduced range, while lower data rates provide longer range at the expense of throughput.

3. Transmission Power:

- Adjusting the transmission power of ESPNow devices can significantly impact communication range. Higher transmission power levels increase the range but may also lead to increased power consumption and potential interference.

4. Payload Size:

- The size of the payload transmitted via ESPNow packets can be customized to accommodate different data requirements. Smaller payloads result in shorter transmission times, while larger payloads allow for more data to be transmitted per packet.

5. Broadcast Mode:

- ESPNow supports both unicast and broadcast modes. Broadcast mode enables the simultaneous transmission of data to multiple devices, which can be useful for scenarios requiring multicast communication.

6. Encryption:

- ESPNow provides support for data encryption to ensure the security and integrity of transmitted data. Enabling encryption adds a layer of protection against unauthorized access and tampering.

7. Time Synchronization:

- Synchronizing the time between ESPNow devices can improve the efficiency and reliability of communication, particularly in applications requiring coordinated actions or timing-sensitive operations.

8. Retry Mechanism:

- Implementing a retry mechanism in ESPNow communication protocols can enhance reliability by automatically retransmitting packets in case of transmission errors or packet loss.

By carefully configuring these parameters based on specific application requirements and environmental conditions, we can optimize the performance of ESPNow-based communication systems to achieve the desired range, data rate, and reliability.

Software Research

Introduction:

The Esp32, a versatile single-board computer, has captured the imagination of hobbyists, students, and professionals alike. Its affordability and flexibility make it an ideal platform for various projects. One common application is interfacing with sensors and displaying their data through a graphical user interface (GUI). This write-up explores the process of creating a GUI on a Esp32 to showcase sensor readings.

Objective:

The primary objective of this project is to design a user-friendly GUI that can present data from electronic sensors in a visually appealing and easily interpretable manner. By doing so, users can monitor sensor data in real-time, facilitating data-driven decisions and enhancing the user experience.

Hardware and Software Requirements:

1. Esp32: The heart of the project, the Esp32, serves as both the data acquisition device and the platform for GUI development.
2. Python Programming: Python, a versatile and easy-to-learn programming language, is used for both sensor data collection and GUI development.
3. ReactJS: React is an open-source JavaScript library developed by Facebook for building user interfaces that are highly dynamic and responsive.
4. MongoDB: MongoDB is a NoSQL database system known for its flexibility and scalability in handling unstructured and semi-structured data.

Reasons and Alternatives:

1. **Esp32:**
    a. Reasons:
        i. Affordability and Accessibility: Esp32 is cost-effective and readily available, making it an excellent choice for hobbyists, students, and small-scale IoT projects.
        ii. Compact Form Factor: Its compact size and low power consumption make it suitable for embedded systems and IoT applications.
        iii. GPIO Pins: Esp32 offers GPIO (General Purpose Input/Output) pins for interfacing with electronic sensors and components.
    b. Alternatives:
        i. Arduino: Arduino boards are another popular choice for hardware interfacing, especially for simpler sensor-based projects.

2. **Python Programming:**
    a. Reasons:
        i. Extensive Libraries: Python has numerous libraries and frameworks for a wide range of applications, including sensor data collection, web development, and data analysis.
        ii. Cross-Platform: Python runs on various platforms, including Esp32, Windows, macOS, and Linux.
    b. Alternatives:
        i. JavaScript (Node.js): Node.js is suitable for server-side development and real-time applications.
        ii. C/C++: These languages offer low-level control and are often used in embedded systems development.
        iii. Java: Java is commonly used in Android app development and enterprise applications.
3. **Node.js:**
    a. Reasons:
        i. JavaScript on the Server: Node.js allows developers to use JavaScript for both front-end (React) and back-end (server) development, reducing the need for context switching.
        ii. Large Package Ecosystem: Node.js has a rich ecosystem of packages and libraries available through npm (Node Package Manager).
        iii. Scalability: It is well-suited for building scalable, real-time applications, such as web servers and chat applications.
    b. Alternatives:
        i. Python (Flask/Django): Python is an alternative for building web backends, especially in cases where integration with Python-based libraries is required.
        ii. Java (Spring Boot): Java is a robust choice for building enterprise-level applications.
4. **MongoDB:**
    a. Reasons:
        i. Schema-less: MongoDB's flexible schema allows easy adaptation to changing data structures, making it suitable for unstructured or semi-structured data.
        ii. Horizontal Scalability: It can scale horizontally, distributing data across multiple servers for improved performance and fault tolerance.
        iii. JSON-like Data Storage: Data in MongoDB is stored in BSON format (Binary JSON), which aligns well with modern development practices.
        iv. Document-Oriented: MongoDB's document-based data model simplifies data representation and retrieval.
    b. Alternatives:
        i. SQL Databases (e.g., MySQL, PostgreSQL): For structured data and transactions, traditional SQL databases may be preferred.
        ii. Firebase: Firebase is a cloud-based platform that includes a NoSQL database and various other services, suitable for mobile and web app development.

5. **React.js:**
    a. Reasons:
        i. Component-Based: React's component-based architecture encourages code reusability and maintainability.
        ii. Large Ecosystem: React has a vast ecosystem of libraries and tools, including state management (Redux, Mobx) and routing (React Router).
        iii. Virtual DOM: React's virtual DOM efficiently updates only the parts of the UI that have changed, improving performance in dynamic applications.
    b. Alternatives:
        i. Angular: Angular is a comprehensive framework by Google for building web applications with a focus on dependency injection and a full-featured ecosystem.
        ii. Vue.js: Vue.js is a progressive JavaScript framework with a smaller learning curve and is known for its ease of integration into existing projects.

Step-by-Step Process

1. Set Up Your Esp32:

- Ensure that your Esp32 is set up and running with a compatible operating system (e.g., Windows 11).
- Make sure you have Node.js and npm (Node Package Manager) installed on your Esp32 to run the Node.js server for your React application.

2. Install MongoDB on Esp32:

- Install MongoDB on your Esp32. You can follow the official MongoDB installation guide for Esp32 or use community-supported packages.
- Create a database and collection in MongoDB where you will store your sensor data.

3. Create a React App:

- Set up a new React application using create-react-app or your preferred React project setup.
- In your React app, you'll create components for the GUI elements you want to display.

4. Build the GUI:

- Design your GUI using React components and libraries like Material-UI, Bootstrap, or any other UI framework of your choice.
- Create forms, charts, or other visual elements to display sensor data.
- Implement features like data input forms for user interaction.

5. Set Up a Node.js Server:

- Create a Node.js server to serve your React app and handle interactions with the Esp32 and MongoDB.
- You can use Express.js or other Node.js frameworks for building your server.

6. Establish Communication with Esp32:

- Set up communication between your React app running on a client machine and the Esp32.
- You can use HTTP requests, WebSockets, or MQTT (Message Queuing Telemetry Transport) for real-time data transfer.

7. Store Data in MongoDB:

- Write API endpoints on your Node.js server to receive data from the Esp32 and store it in MongoDB.
- Utilize a MongoDB driver for Node.js to interact with the database.

8. Display Sensor Data in the GUI:

- Fetch data from MongoDB through your Node.js server and display it in your React GUI.
- Use asynchronous requests (e.g., Axios or Fetch) to retrieve data from the server.

9. Real-Time Updates (Optional):

- If you want real-time updates in your React app, consider using WebSockets or a library like Socket.io for bi-directional communication between the client and server.

10. Testing and Debugging:

- Thoroughly test your React app, server, and database interactions. Debug and refine your code as needed.

11. Deployment:

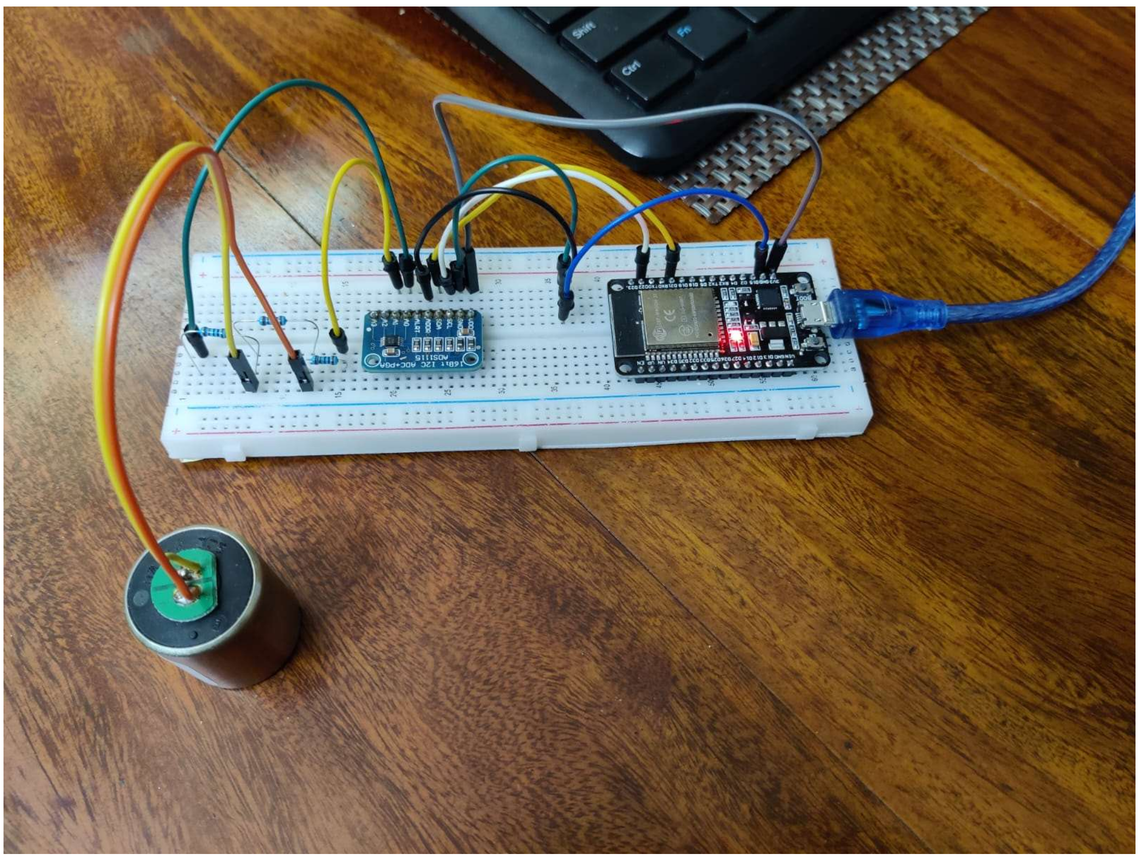
- Deploy your React app and Node.js server. You can use platforms like Heroku, AWS, or a VPS (Virtual Private Server) to host your application.

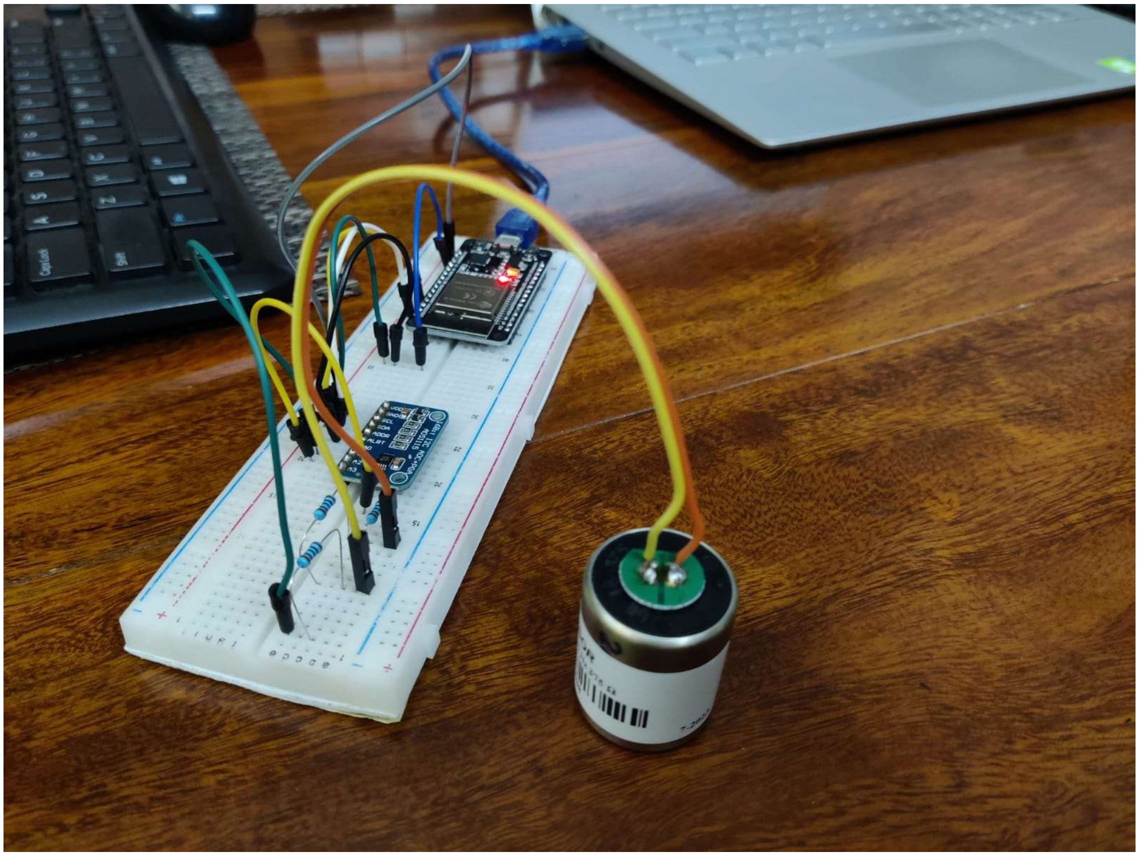
Conclusion:

Creating a web-based GUI using React, along with a Esp32 storing data in MongoDB, combines modern web development with IoT capabilities. This project enables users to interact with and visualize sensor data through a user-friendly interface accessible from anywhere with an internet connection. It showcases the power of web technologies and Esp32 for building practical IoT applications with real-world applications in mind.

Final Working Connections

Here's where everything comes together. After all the planning and testing, we've got our circuit working just right on the breadboard. It's the result of careful steps, combining different parts like microcontrollers, sensors, and communication modules. Each connection is like a piece of a puzzle fitting perfectly into place. Now, as we snap some pictures of our final setup, we're capturing the essence of our project—the hard work, the problem-solving, and the innovation. These images tell the story of what we've accomplished, turning ideas into reality in the world of embedded systems and IoT.





Transmitter Codes

Here, we detail the programming behind our transmitter unit. These lines of code are the backbone of our system, instructing the microcontroller on how to gather data from sensors, process it, and transmit it wirelessly. They represent our understanding of the problem at hand and how we've translated it into actionable commands for the hardware. With each line meticulously written and tested, these codes embody our expertise in software development and signal the progress towards our project's success.

Final_Sender | Arduino 1.8.19

File Edit Sketch Tools Help

Final_Sender

```
#include <Wire.h>
#include <Adafruit_ADS1X15.h>
#include <cstdint>

Adafruit_ADS1115 ads;

#include <esp_now.h>
#include <WiFi.h>

#define CHANNEL 1

esp_now_peer_info_t slave;

uint8_t data1 = 0;
uint8_t data2 = 0;

// Set the desired number of readings per second
int readingsPerSecond = 50; // Adjust this value as needed
int delayBetweenReadings = 1000 / readingsPerSecond;


// Function to convert unsigned 16-bit data to an array of unsigned 8-bit numbers
void uint16_to_uint8_array(uint16_t data, uint8_t *array) {
    array[0] = (uint8_t)(data >> 8);   // High byte
    array[1] = (uint8_t)data;          // Low byte
}

uint8_t array_data[2];
uint16_t answer;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  Serial.println("Hello, I am working fine");

  if (!ads.begin()) {
    Serial.println("Failed to initialize ADS.");
    while (1);
  }
  WiFi.mode(WIFI_STA);
  esp_now_init();
```

```
  esp_now_register_send_cb(OnDataSent);
  ScanForSlave();
  esp_now_add_peer(&slave);
}

void loop() {
  // put your main code here, to run repeatedly:
  int16_t results;
  results = ads.readADC_Differential_0_1();
  answer = results + 32768;

  uint16_to_uint8_array(answer, array_data);

  data1 = array_data[0];
  data2 = array_data[1];


  esp_now_send(slave.peer_addr, &data1, sizeof(data1));
  //delay(10);
  esp_now_send(slave.peer_addr, &data2, sizeof(data2));

  //Serial.print(data1);
  //Serial.print(",");
  //Serial.print(data2);
  //Serial.print(",");
  Serial.println(results);
  delay(delayBetweenReadings);

}




void ScanForSlave() {
  int8_t scanResults = WiFi.scanNetworks();

  for(int i = 0; i < scanResults; ++i) {
    String SSID = WiFi.SSID(i);
    String BSSIDstr = WiFi.BSSIDstr(i);

    if (SSID.indexOf("RX") == 0){
```

```

      int mac[6];
      if( 6 == sscanf(BSSIDstr.c_str(), "%x:%x:%x:%x:%x:%x", &mac[0], &mac[1], &mac[2], &mac[3], &mac[4], &mac[5] )){
         for (int ii = 0; ii < 6; ++ii ){
          slave.peer_addr[ii] = (uint8_t) mac[ii];
        }
      }

      slave.channel = CHANNEL;
      slave.encrypt = 0;
      break;
    }
  }
}


void OnDataSent(const uint8_t *mac_addr, esp_now_send_status_t status){
  //Serial.println(data1);
  //Serial.println(data2);
}
```

Receiver Codes

In this section, we unveil the programming logic behind our receiver unit. These codes decode transmitted data, enabling the receiver to interpret and respond to incoming information. Crafted with precision, they seamlessly integrate with our hardware setup, facilitating smooth communication between devices. Exploring these codes illuminates their pivotal role in our system's functionality, showcasing our adeptness in software development within embedded systems.

Final_Receiver | Arduino 1.8.19

File Edit Sketch Tools Help

Final_Receiver

```
#include <esp_now.h>
#include <WiFi.h>
#include <cstdint>
#include <Wire.h>

#define CHANNEL 1


uint8_t storage[2];
int positionnumber = 0;


// Set the desired value of sensitivity ranging from 1(Min.) to 6(Max.)
int Sensitivity = 6;
float multiplier;



void setup() {
  Serial.begin(115200);

  WiFi.mode(WIFI_AP);
  WiFi.softAP("RX_1", "RX_1_Password", CHANNEL, 0);

  esp_now_init();
  esp_now_register_recv_cb(OnDataRecv);

  Serial.println("Hello, I am working fine");

  if (Sensitivity == 1) {
    multiplier = 0.0078125F;
  } else if (Sensitivity == 2) {
    multiplier = 0.015625F;
  } else if (Sensitivity == 3) {
    multiplier = 0.03125F;
  } else if (Sensitivity == 4) {
    multiplier = 0.0625F;
  } else if (Sensitivity == 5) {
    multiplier = 0.125F;
  } else if (Sensitivity == 6) {
    multiplier = 0.1875F;
```

Done Saving.

```
  } else {
    Serial.println("Invalid Sensitivity level");
  }
}

void loop() {

}


void OnDataRecv(const uint8_t *mac_addr, const uint8_t *data, int data_len){
  if (positionnumber == 0){
    storage[positionnumber] = *data;
    positionnumber++;
    }
  else if (positionnumber == 1){
    storage[positionnumber] = *data;
    positionnumber++;
   }
  else {
    //Serial.print(storage[0]);
    //Serial.print(",");
    //Serial.print(storage[1]);
   // Serial.print(",");
    uint16_t converted_data = uint8_array_to_uint16(storage);
    int16_t int16_data = convert_uint16_to_int16(converted_data);
    int16_t result = int16_data + 32768;
    //Serial.print(converted_data);
    //Serial.print(",");
    //Serial.println(result);
   // Serial.print(",");
    Serial.println(result * multiplier);

    positionnumber = 0;
    storage[positionnumber] = *data;
    positionnumber++;

  }

}

```

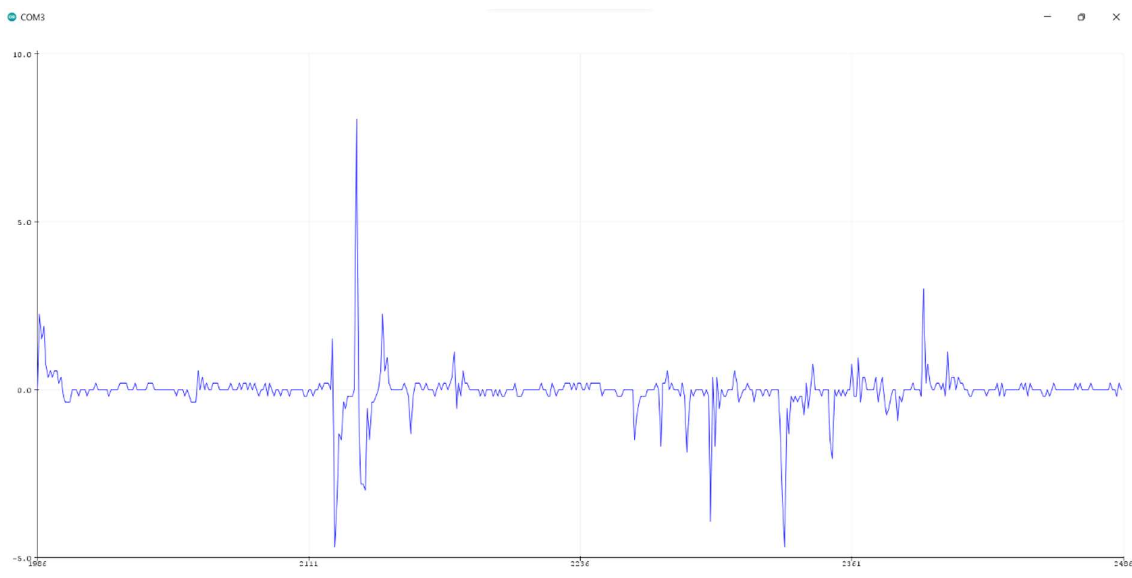
Done Saving.

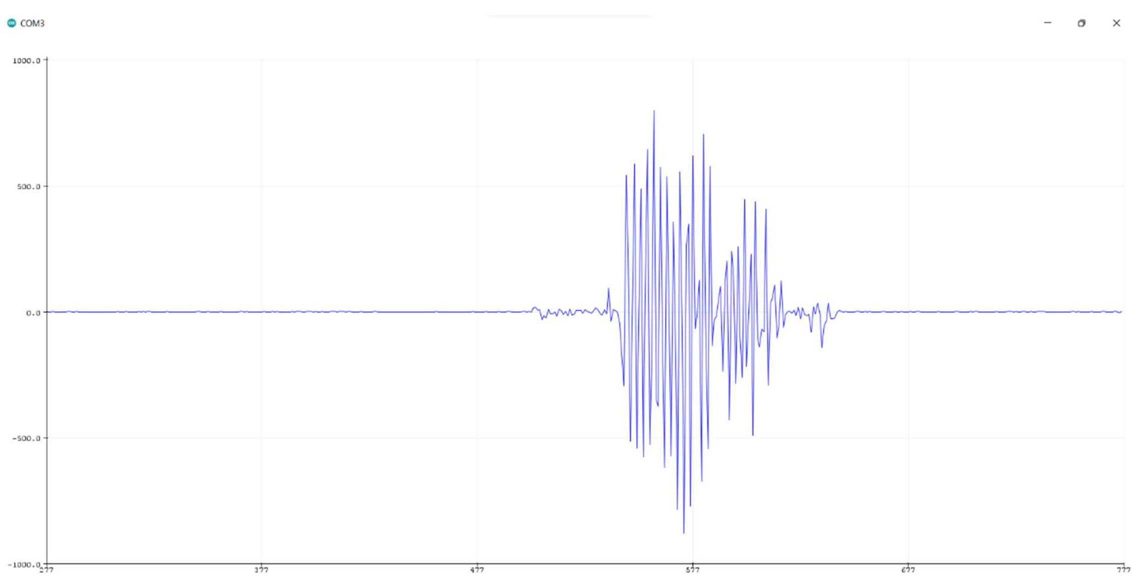
```

int16_t convert_uint16_to_int16(uint16_t uint16_data) {
  int16_t int16_data = static_cast<int16_t>(uint16_data);
  return int16_data;
}


// Function to convert an array of unsigned 8-bit numbers back to unsigned 16-bit data
uint16_t uint8_array_to_uint16(const uint8_t *array) {
    return ((uint16_t)array[0] << 8) | array[1];
}
```

Done Saving.

Received Data Graphs

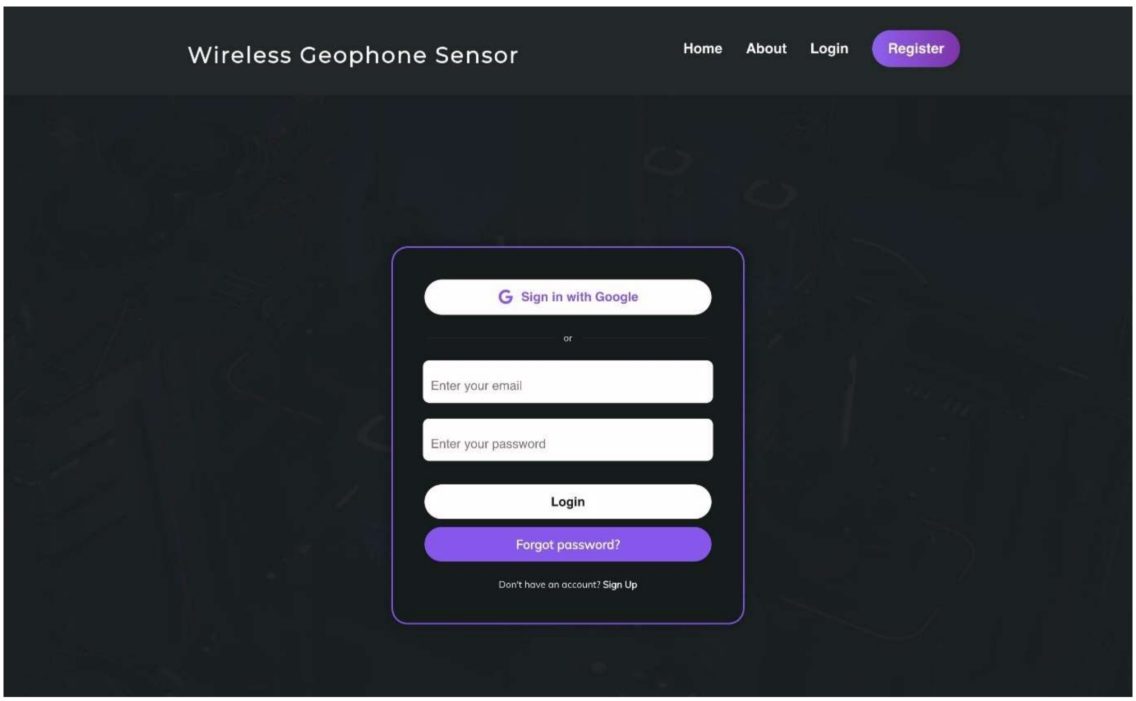
Under this heading, we present visual representations of the data received by our system. These graphs offer insights into the performance and behavior of our device, showcasing trends, patterns, and anomalies in the transmitted data. Through these visualizations, we gain a deeper understanding of how our system interacts with its environment and how it responds to various stimuli. Each graph serves as a window into the functionality and effectiveness of our project, providing valuable feedback for further refinement and optimization.



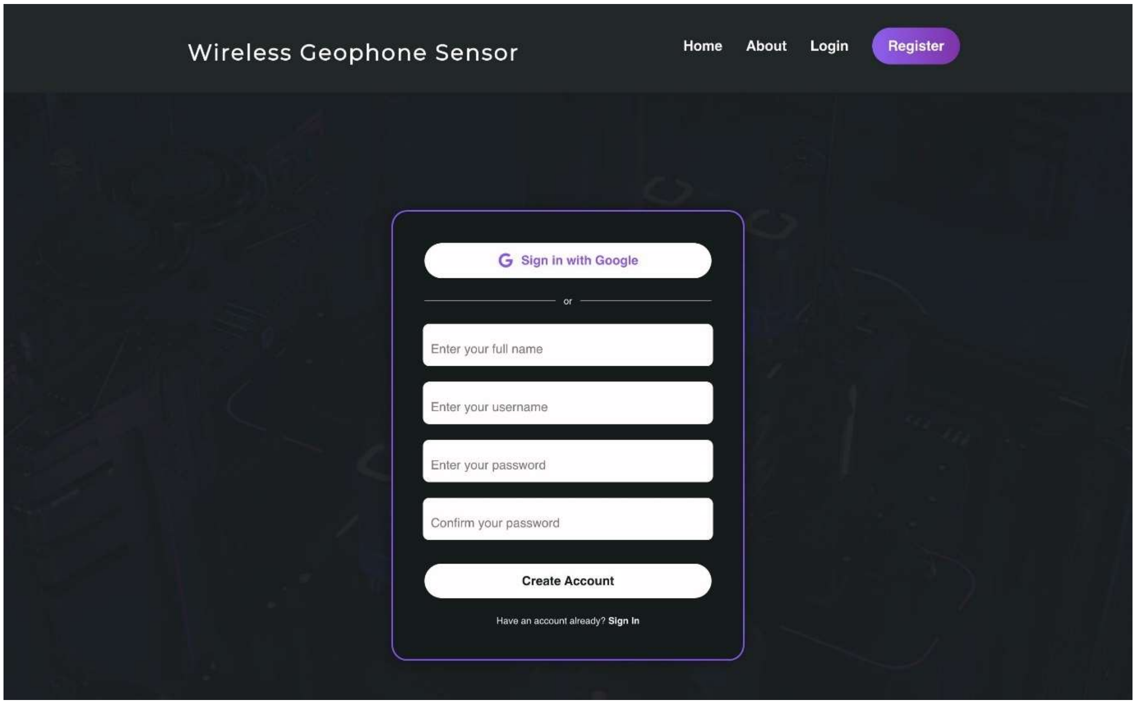


Software Frontend (GUI)

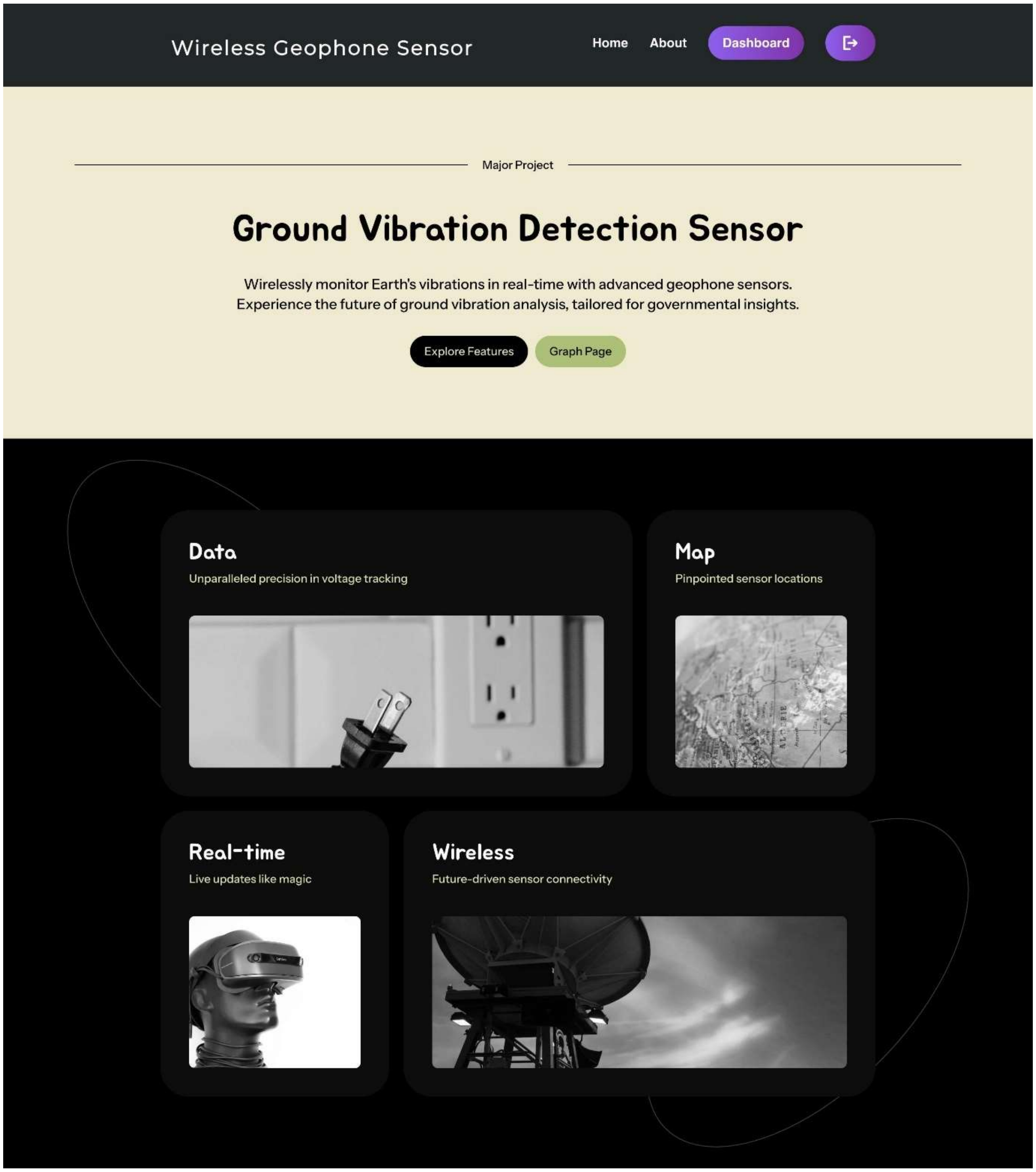
In this section, we introduce the graphical user interface (GUI) that serves as the frontend of our software application. Designed with user-friendliness in mind, the GUI provides a visually intuitive platform for users to interact with our system. Through this interface, users can easily access and control various functionalities, visualize real-time data, and configure settings as needed. The GUI enhances the overall user experience by simplifying complex operations and presenting information in a clear and organized manner. As we delve into the details of our software frontend, we highlight the efforts invested in creating a seamless and engaging interface that enhances the usability and accessibility of our system.



In our software frontend GUI, we've implemented a secure login page to ensure controlled access and protect sensitive information. This login feature serves as the gateway to our system, requiring authorized users to authenticate themselves before gaining access to its functionalities. Through robust encryption protocols and user authentication mechanisms, we've fortified the login page against unauthorized access and potential security breaches. By incorporating this secure login functionality, we prioritize the confidentiality and integrity of our system's data, instilling confidence in users regarding the safety of their information. As we navigate through the intricacies of our GUI, the presence of this secure login page underscores our commitment to maintaining the highest standards of security and user privacy.



The landing page of our website serves as the central hub where users can access various sections tailored to their needs and preferences. Here, we offer a dynamic array of features, each designed to provide valuable insights and enhance the user experience. Among these sections, users can explore live data plots, offering real-time visualization of incoming data streams, enabling them to monitor trends and make informed decisions on the fly. Additionally, the received data section presents a comprehensive overview of all data received by the system, organized and displayed for easy reference and analysis. Furthermore, we provide a captivating representation on the map, allowing users to geographically visualize data points and patterns, facilitating spatial analysis and contextual understanding. Through these diverse sections, our landing page offers a holistic view of our system's capabilities, empowering users with the tools they need to derive meaningful insights and drive informed actions.



Wireless Geophone Sensor
Home
About
Dashboard
Major Project
Ground Vibration Detection Sensor
Wirelessly monitor Earth's vibrations in real-time with advanced geophone sensors.
Experience the future of ground vibration analysis, tailored for governmental insights.
Explore Features
Graph Page
Data
Unparalleled precision in voltage tracking
Map
Pinpointed sensor locations
Real-time
Live updates like magic
Wireless
Future-driven sensor connectivity

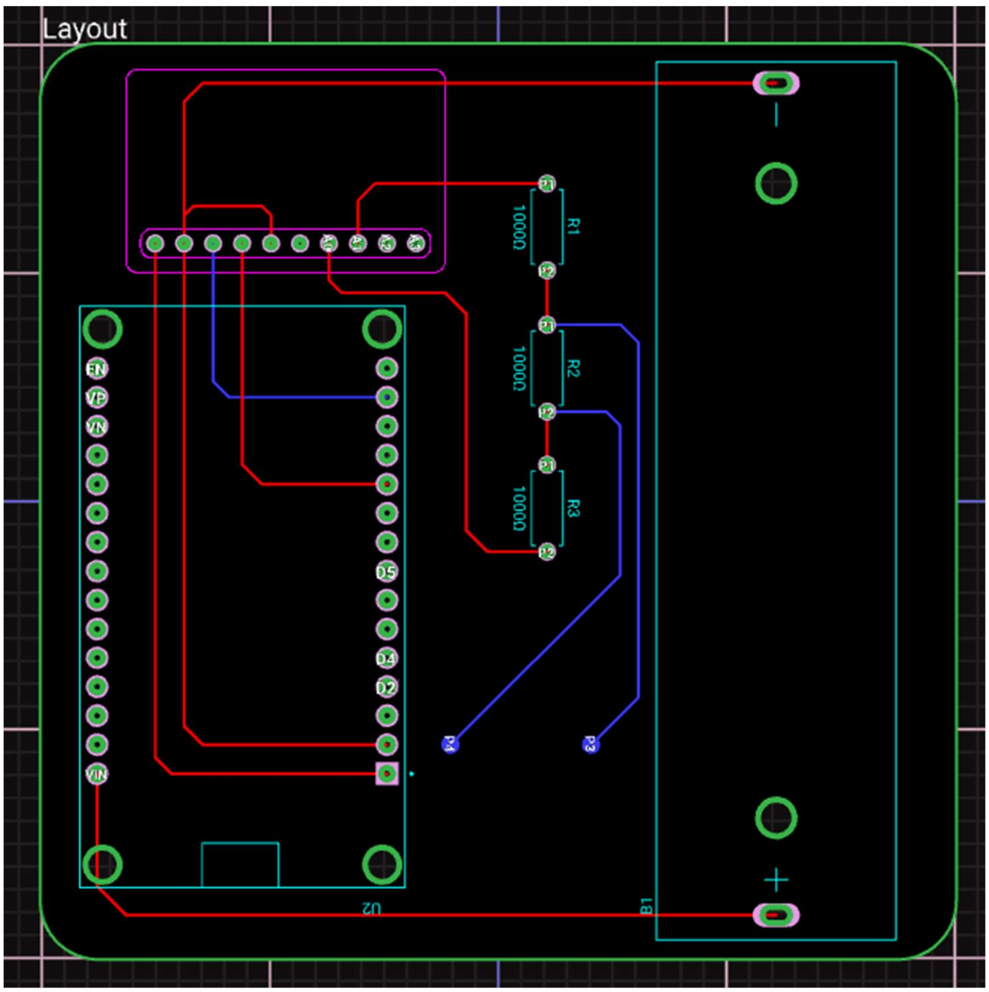
Software Backend Build

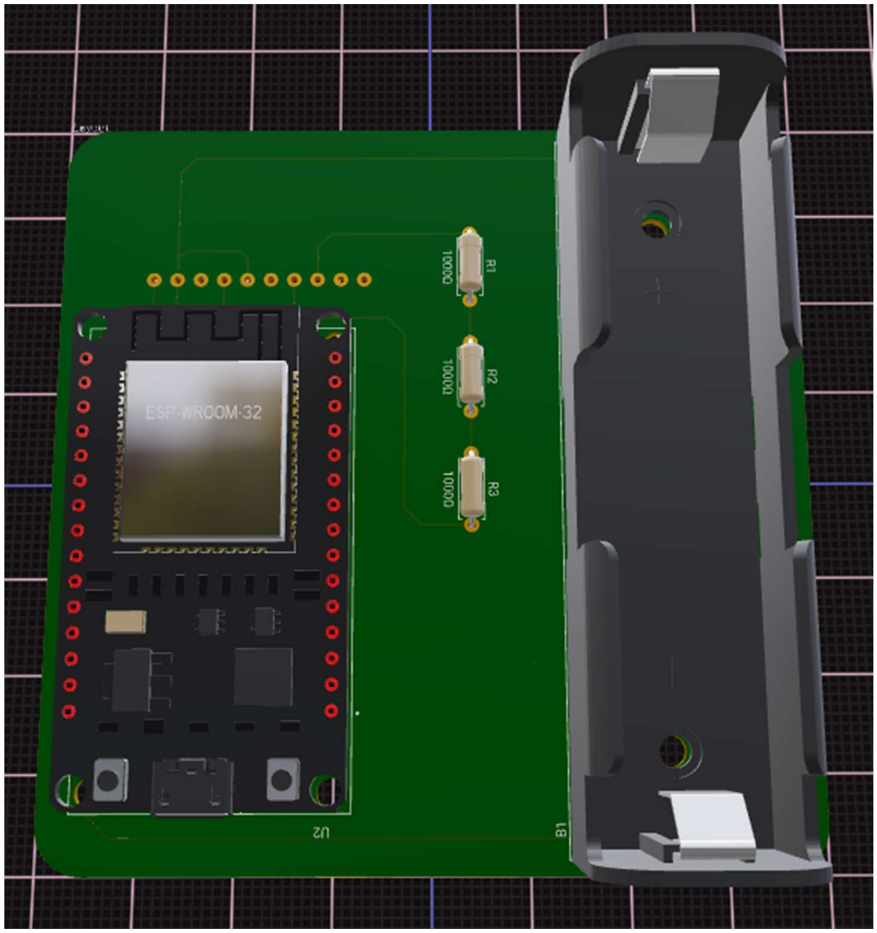
Under this heading, we delve into the architecture and implementation of the backend code that powers our GUI application. This backend build serves as the engine behind the graphical user interface, handling data processing, business logic, and system interactions. Through meticulous coding and optimization, we've crafted a robust and scalable backend infrastructure that seamlessly integrates with our frontend interface. The backend code ensures smooth communication with hardware components, data storage, and retrieval, and facilitates secure user authentication and authorization processes. As we explore the intricacies of our backend build, we unveil the craftsmanship and expertise invested in creating a reliable and efficient foundation for our GUI application, laying the groundwork for a seamless user experience.

```
const express = require('express');
const mongoose = require('mongoose');
const bodyParser = require('body-parser');
const cookieParser = require('cookie-parser');
const cors = require('cors');
const session = require('express-session');
const serverless = require('serverless-http');
const path = require('path');
require('dotenv').config({ path: path.resolve(__dirname, './.env') });

const app = express();

const allowedOrigins =  'http://localhost:3000', 'http://localhost:8000'];
                      [
app.use(
    cors({
        origin(origin, callback) {
            if  !origin || allowedOrigins.indexOf(origin) !== -1) {
              ( callback(null, true);
            } else
                ca{lback(new Error('Not allowed by CORS'));
            }
        },
    })
);

const passport = require('passport');
require('./middleware/passport');

const authRoutes = require('./routes/authRoutes');
const usersRoutes = require('./routes/userRoutes');

// Set up body parser middleware
app.use(bodyParser.urlencoded({ extended: true }));
app.use(bodyParser.json());

// Set up cookie parser middleware
app.use(cookieParser());

// call config
const config = require('./config');
const { errorLoggerMiddleware, logger } = require('./middleware/logger');

// connect to mongodb
mongoose.connect(config.dbConnectionString, () =>
    logger.info('mongoDb server has been connecte{!');
});

// Set up Passport middleware dependency
app.use(session({ secret: 'cats', resave: false, saveUninitialized: true }));
app.use(passport.initialize());
app.use(passport.session());

// Use routes
// app.use("/", mainRoutes);
app.use('/auth', authRoutes);
app.use('/users', usersRoutes);

// Set up error handling middleware
app.use(errorLoggerMiddleware);

// Start server
const PORT = config?.server?.port || 8000;
app.listen(PORT, () =>
    logger.info(`Serve{ running on port ${PORT}`);
});

// Export as serverless handler if needed
module.exports.handler = serverless(app);
```

PCB Design

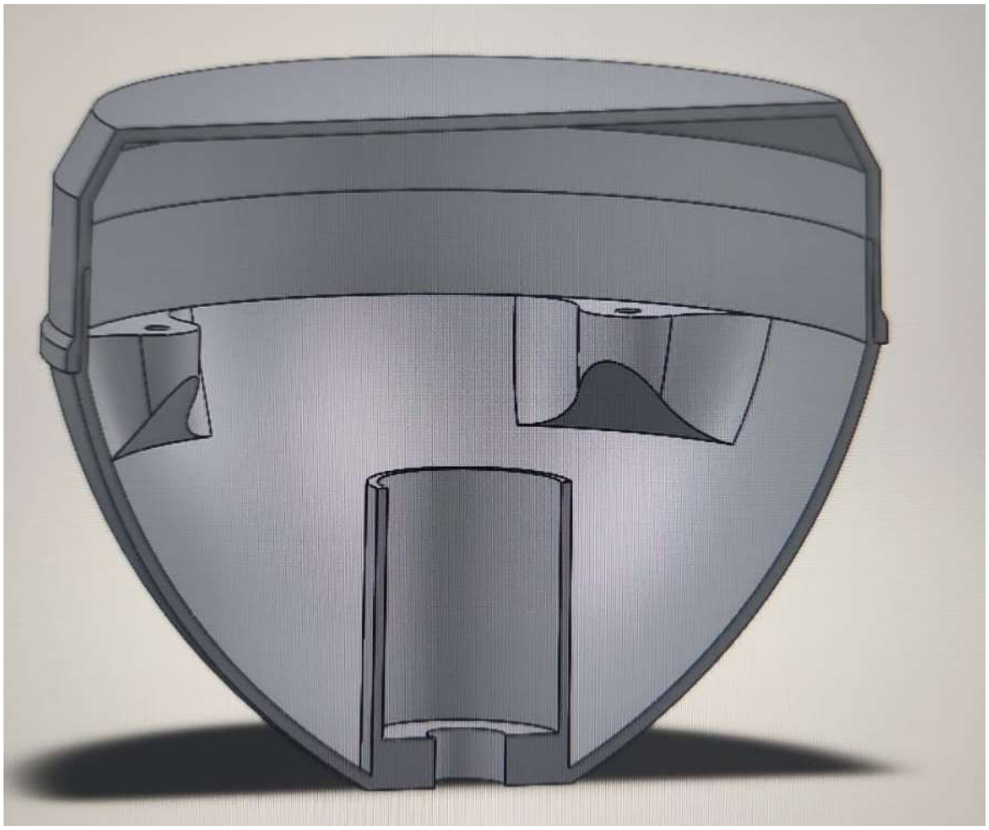
In this section, we dive into the intricacies of our printed circuit board (PCB) design, elucidating the meticulous process of translating our schematic diagrams into tangible hardware. Through careful consideration of component placement, signal routing, and electrical considerations, we've crafted a PCB layout that optimizes performance, minimizes interference, and ensures reliability. Each trace and component footprint is strategically placed to maximize efficiency and functionality while adhering to industry standards and best practices. As we delve into the details of our PCB design, we unveil the culmination of our engineering expertise and innovation, culminating in a compact and elegant solution that serves as the backbone of our electronic system.





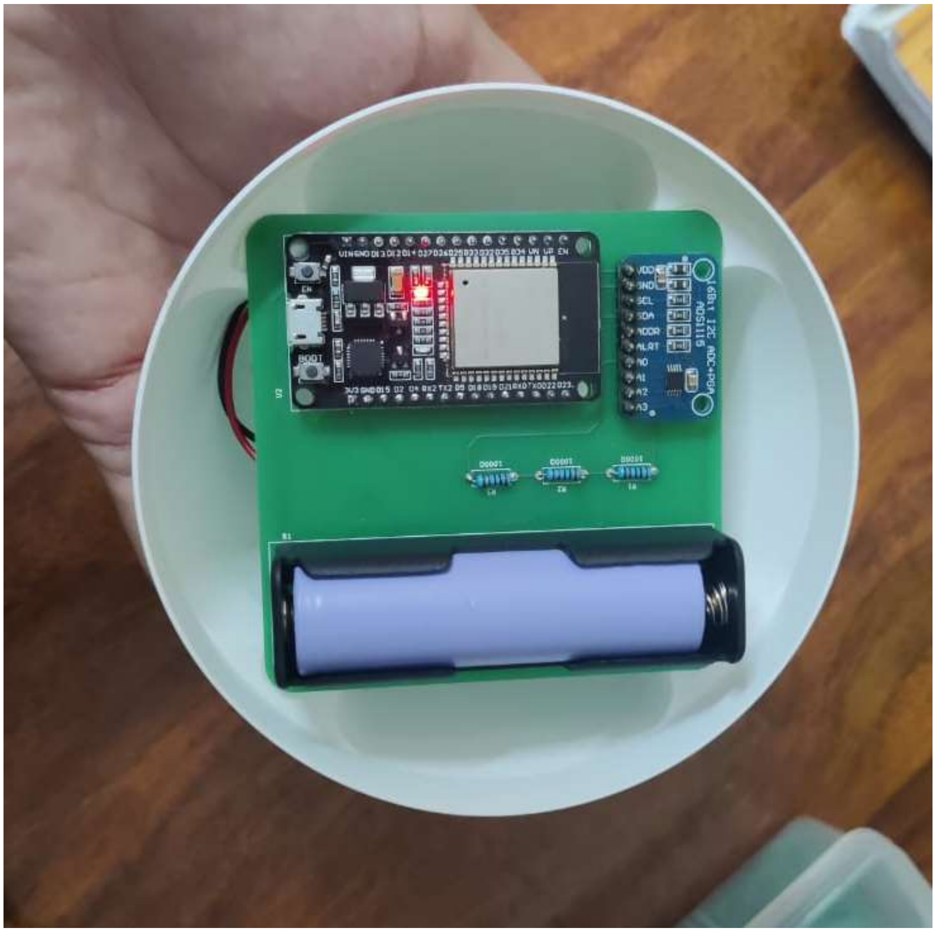
3D Case Design

Under this heading, we explore the process of conceptualizing and creating the physical enclosure for our electronic system. Through 3D modeling software and prototyping techniques, we have meticulously crafted a case design that not only houses our components securely but also enhances the aesthetic appeal and usability of our device. Each aspect of the design, from form factor to ventilation and accessibility, has been carefully considered to ensure optimal functionality and user experience. As we delve into the intricacies of our 3D case design, we highlight the fusion of creativity and functionality, resulting in a sleek and ergonomic enclosure that complements the innovative technology housed within.



Sensor Final Design

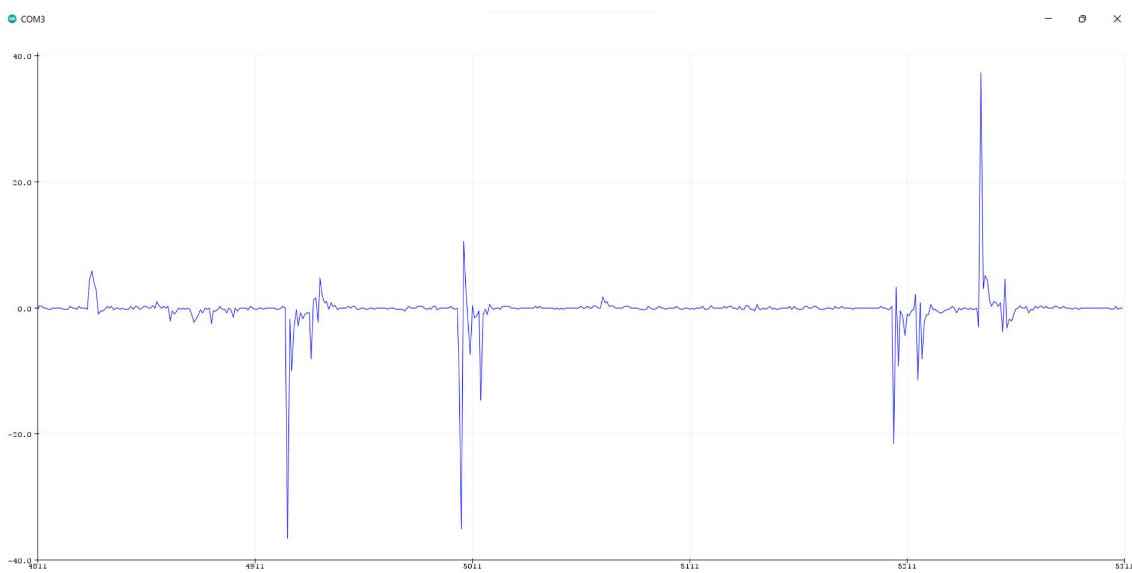
This section showcases the culmination of our sensor design efforts, presenting the fully assembled sensor unit in its completed form. Through meticulous attention to detail and iterative refinement, we've achieved a sensor design that encapsulates both functionality and aesthetics. The pictures featured here capture every aspect of the sensor, from the intricacies of the circuitry to the sleek exterior casing, showcasing the seamless integration of hardware components. As we unveil the final sensor design, we celebrate the synergy of engineering precision and design elegance, underscoring our commitment to delivering a sensor solution that excels in both performance and visual appeal.



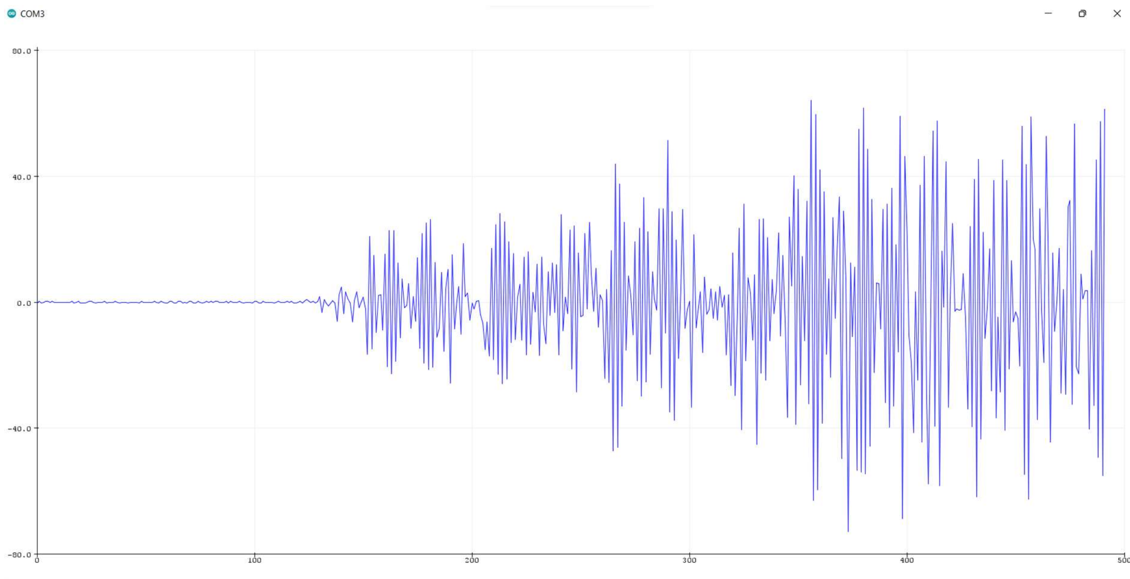
Ground Testing

A Human Walking

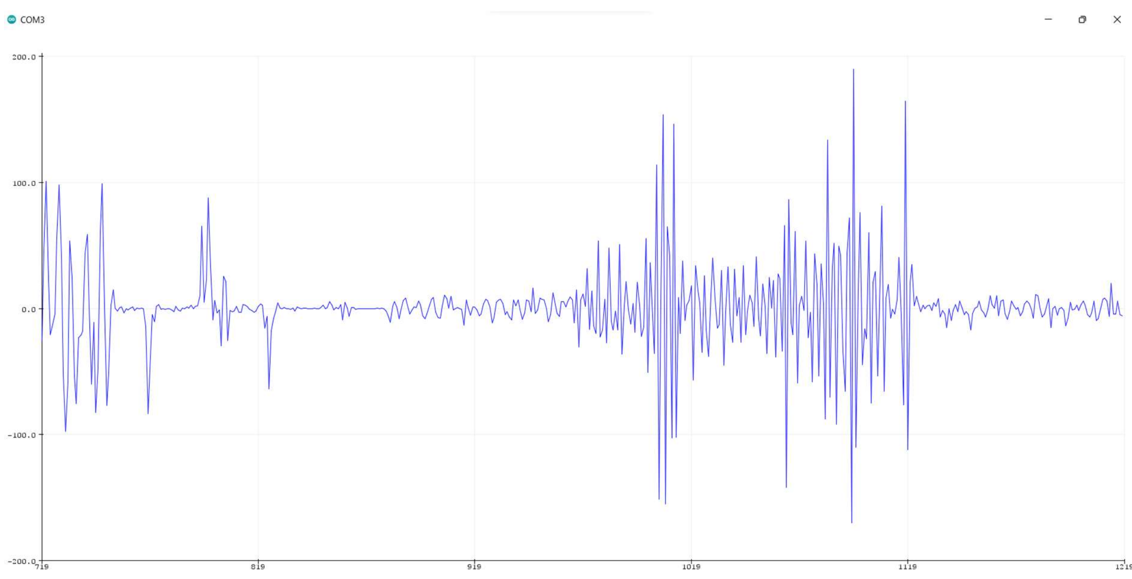
A Human Jumping



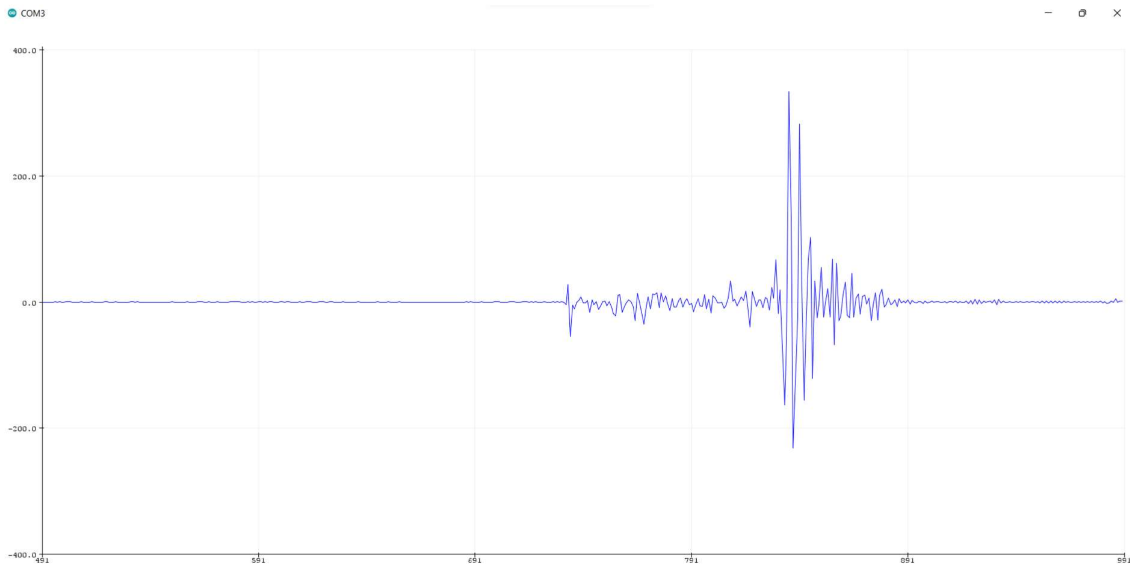
A Bike Passing



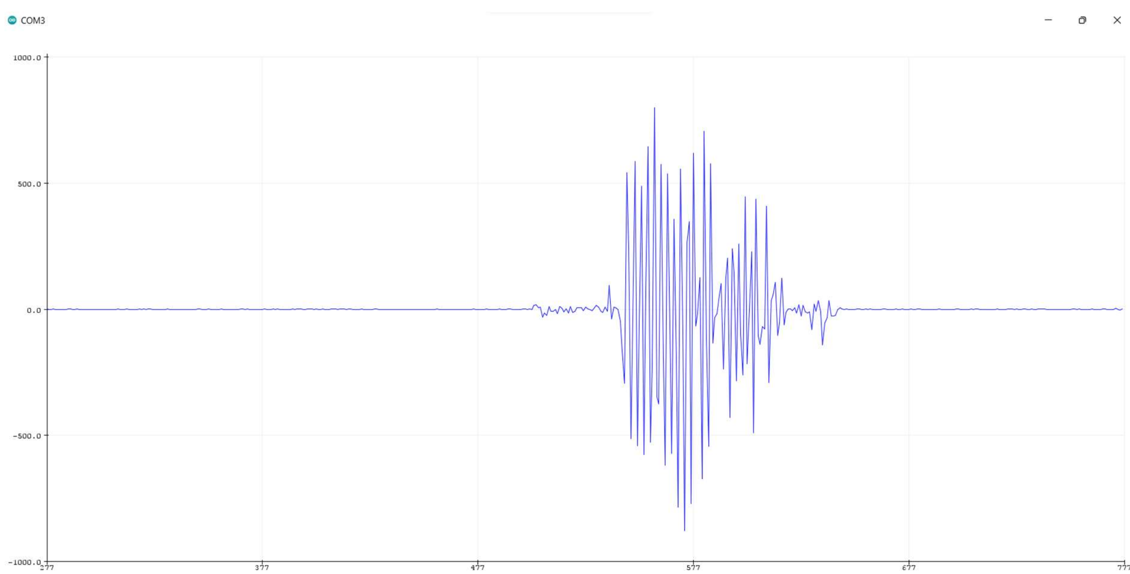
A Car Passing



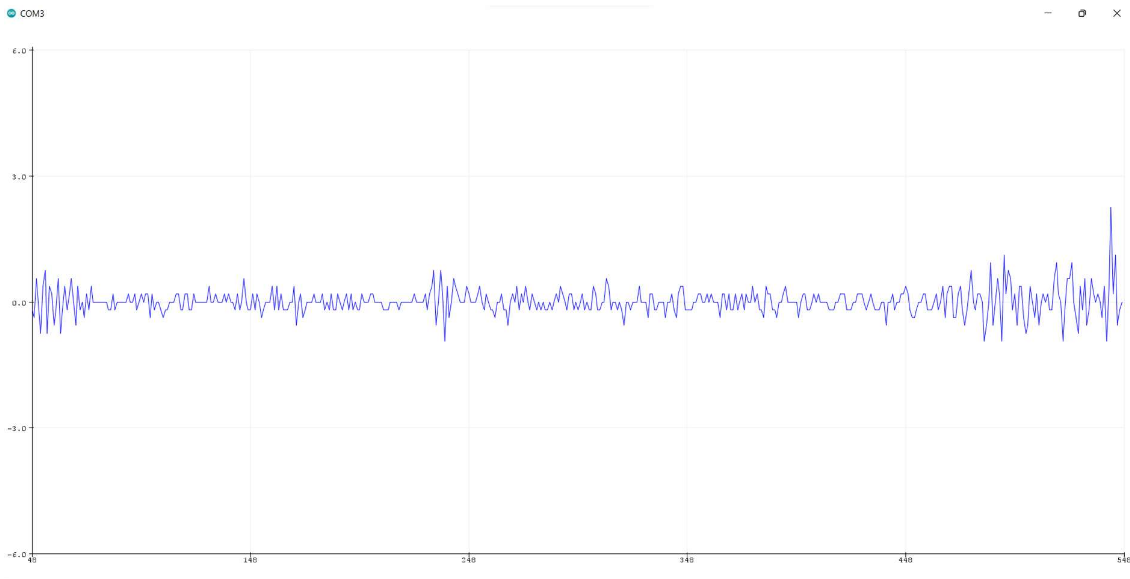
A Small Firecracker



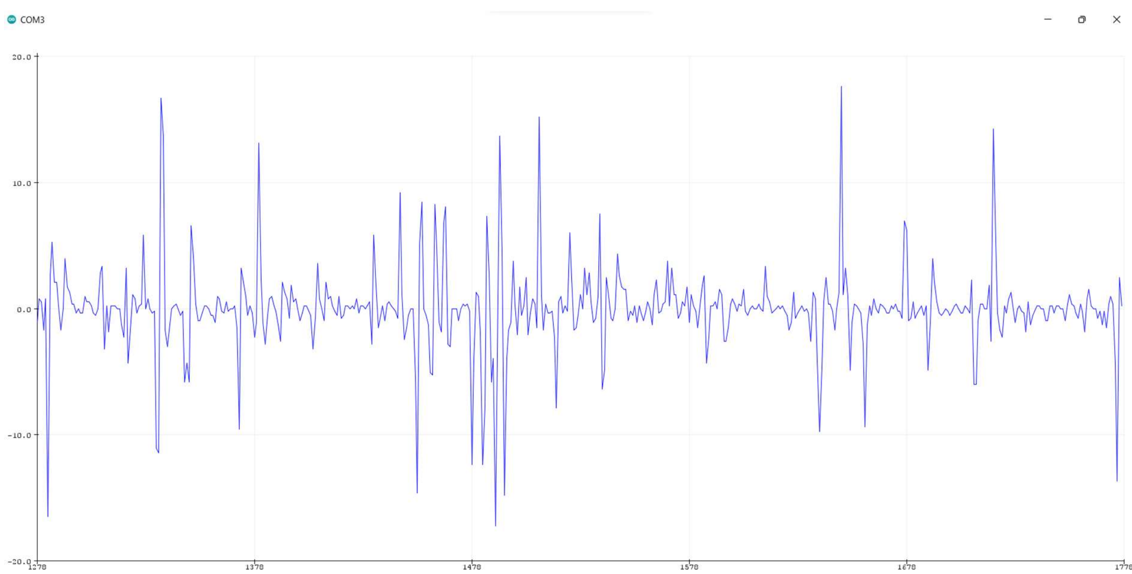
A Big Firecracker



A Calm Street



A Buisy Street



Conclusion

Conclusion

The Wireless Ground Vibration Sensor Project represents a significant leap forward in the realm of ground vibration monitoring technology. Ground vibrations, a critical aspect of various industrial activities, demand meticulous attention for ensuring structural integrity, environmental sustainability, and human safety. Traditional wired seismic sensors, while effective, are burdened by limitations in terms of setup complexity, maintenance, scalability, and real-time data access.

Our project addresses these challenges head-on, introducing a Wireless Ground Vibration Sensor System that harnesses cutting-edge technologies to revolutionize the field. By incorporating advanced geophone sensors, analog-to-digital conversion, and 2.4Ghz wireless communication, the wireless system promises enhanced accuracy, scalability, and real-time data accessibility. The geophone sensors provide superior data quality, overcoming the limitations associated with alternative sensor technologies.

The applications of the Wireless Ground Vibration Sensor system are diverse and impactful, ranging from ensuring safety in construction and mining to assessing the impact of explosive events on structures. The system's ability to contribute to environmental monitoring, seismic research, and broader industrial applications showcases its versatility and potential.

The project's objectives align with a broader vision of offering a cost-effective, efficient, and scalable solution to ground vibration monitoring. By eliminating the constraints of physical connectivity and embracing wireless technology, the project aims to redefine industry standards and usher in a new era of innovation.

Informed by the research conducted on ground vibrations, types of vibrations, and traditional sensing methods, the project methodology meticulously integrates geophone sensors, ADC, ESP32 microcontrollers, and 2.4Ghz wireless technology. This combination ensures not only precise measurement and analysis of ground vibrations but also seamless wireless access to the collected data, promoting collaboration and timely decision-making.

As the project progresses, it holds the promise of contributing significantly to the advancement of ground vibration monitoring technology. The envisioned system has the potential to become a benchmark for efficiency, scalability, and cost-effectiveness, setting new standards in the industry. The Wireless Ground Vibration Sensor Project represents a transformative endeavour that not only addresses current limitations but also anticipates future needs in the dynamic landscape of ground vibration monitoring.

Future Scope

Future Scope

The Wireless Ground Vibration Sensor Project lays a solid foundation for future advancements and extensions, opening avenues for innovation and refinement in ground vibration monitoring technology. The project's future scope encompasses various aspects, including technological enhancements, expanded applications, and integration with emerging trends. Here are key areas for future exploration:

1. Sensor Technology Advancements:

- Multisensor Integration: Explore the integration of multiple sensor types, such as accelerometers and strain gauges, to provide a more comprehensive understanding of ground vibrations and structural responses.

- Miniaturization: Investigate advancements in sensor miniaturization for increased portability and deployment flexibility, enabling monitoring in challenging terrains and environments.

2. Data Analysis and Machine Learning:

- Advanced Analytics: Implement advanced data analytics techniques, including machine learning algorithms, for more nuanced interpretation of ground vibration data. This could enhance the system's ability to detect patterns, anomalies, and predict potential structural impacts.

- Automated Anomaly Detection: Develop algorithms for automated anomaly detection to promptly identify and alert users to abnormal ground vibration patterns, enabling rapid response to potential safety concerns.

3. IoT Integration and Edge Computing:

- Internet of Things (IoT) Integration: Explore integration with IoT frameworks to enhance connectivity, allowing for seamless integration with other smart infrastructure components and centralized monitoring platforms.

- Edge Computing: Investigate the implementation of edge computing for decentralized data processing, reducing latency and enabling real-time decision-making at the sensor deployment site.

4. Environmental Monitoring Applications:

- Biodiversity Impact Assessment: Extend the system's capabilities to assess the impact of ground vibrations on local flora and fauna, contributing to more comprehensive environmental impact assessments.

- Air and Noise Quality Monitoring: Integrate sensors for monitoring air quality and noise levels in conjunction with ground vibration data to provide a holistic view of the environmental impact of industrial activities.

5. Global Collaboration and Standardization:

- Collaborative Research Networks: Foster collaboration with research institutions, industries, and governmental bodies globally to create shared databases and insights into ground vibrations. This can lead to the establishment of best practices and standards for ground vibration monitoring.

- Regulatory Compliance: Work towards aligning the developed system with international standards and regulations, facilitating its adoption across various industries and geographical regions.

6. Energy Harvesting and Sustainability:

- Energy-Efficient Sensors: Explore the integration of energy-efficient sensor technologies and energy harvesting mechanisms to reduce the environmental impact of sensor deployments and extend their operational life.

- Sustainable Materials: Investigate the use of sustainable and eco-friendly materials in sensor manufacturing to align with broader sustainability goals.

7. Real-Time Visualization and Public Awareness:

- Public-Facing Dashboards: Develop public-facing dashboards or interfaces for real-time visualization of ground vibration data, fostering public awareness and transparency regarding industrial activities' impact on surrounding areas.

- Community Engagement: Implement strategies for community engagement, allowing residents to access and understand ground vibration data, thus promoting a collaborative approach to addressing concerns and mitigating potential impacts.

8. Smart Infrastructure Integration:

- Integration with Smart Cities: Explore integration possibilities with smart city initiatives, where ground vibration data can contribute to the overall intelligence of urban infrastructure, enhancing resilience and sustainability.

- Infrastructure Health Monitoring: Extend the system's capabilities to monitor the health of critical infrastructure, such as bridges and pipelines, by detecting and assessing ground vibrations induced by various factors.

The future scope of the Wireless Ground Vibration Sensor Project is expansive, driven by a commitment to continuous improvement, collaboration, and adaptability to emerging technologies and trends. By embracing these future directions, the project can contribute significantly to advancing ground vibration monitoring and its applications across diverse domains.

Team Members

- Omkar Singh Sandhu 20105004
- Akshat Kumar 20105005
- Agrim Gupta 20105008
- Rahul Sharma 20105015
- Kanwar Ekamjot Singh 20105021

Mentor Dr. Jyoti Kedia

Thank You