IRJIEST 10(2) 2024 14-23
Available online at https://ojs.batstate-u.edu.ph
International Research Journal on Innovations in Engineering, Science and Technology

Development of a wireless sensor node as an earthquake monitoring system for buildings

John Carlo V. Aggari*, Mirasol C. Dilay, Jeffrey S. Sarmiento, John Iszach D. Serbeto
College of Engineering, Batangas State University The National Engineering University, Batangas, Philippines

ABSTRACT
The study aimed to design and develop a Wireless Sensor Node (WSN) to monitor vibration data in buildings. The goal was to create an affordable tool that aligns with the Department of Public Works and Highways efforts to track building conditions before, during, and after earthquakes. It also supports the Building Research and Innovation Development Goals for Engineering SUCs, which fosters collaboration among engineering state universities and colleges across the Philippines to address community challenges through research. The WSN includes essential features like data collection, verification, processing, and cloud storage. The system consists of an accelerometer sensor, a microcontroller, and an uninterruptible power supply for consistent operation. The sensor gathers acceleration data in the time domain, which is then converted to the frequency domain using the Fast Fourier Transform for easier analysis. To implement the system, the WSN was deployed in a building to collect baseline vibration data, which was used to set thresholds for the sensor's algorithms. Initial tests showed that the WSN successfully measured building vibrations, and the data was securely stored in the cloud, where it could be accessed through a website. This study set the foundation for future improvements and wider use of this technology.

Keywords: wireless sensor, earthquake, accelerometer, MEMS, cloud

Received: May 31, 2024 **Revised:** August 26, 2024 **Accepted:** November 15, 2024

1. Introduction

Many natural disasters can happen in every part of the world. Some of these disasters are storms, volcanic eruptions, tornadoes, wildfires, earthquakes, and many more. All these natural disasters can cause thousands, or in some cases millions and billions, in damages—not only in terms of lives lost, but also in the worth of infrastructures. On the good part, some of these natural disasters can be seen coming. Like storms and volcanic eruptions, there are some indications that something is going to happen. With the advancement of technology, some of the things about the disaster can be predicted. When a storm is anticipated to enter our area of responsibility, there are government agencies that track and project the path and the intensity of the storms which can alert and issue a signal level on a certain part of the country. On the other hand, some calamities are unpredictable which makes it more destructive. One example of these natural disasters that can happen without any signs or indication is an earthquake.

An earthquake is any sudden shaking of the ground caused by the passage of seismic waves through the Earth's rocks [1]. There are many records of the occurrence of these natural calamities and evidently, they claimed thousands of lives and caused harm to some monumental property. Natural calamities are harmful during the time of their occurrence but unlike other natural occurring disasters, earthquake after-effects are more dangerous and destructive. To come up with a solution to these problems, data about the building before, during, and after the earthquake is needed. Thus, the idea of monitoring earthquake after-effects on a building emerged.

*Corresponding author
Email address: johncarlo.aggari@g.batstate-u.edu.ph

Monitoring of earthquake shaking in a building could be a key to rising building style. To enhance the stable performance of buildings and infrastructure, engineers want a much better understanding of the existing structure to respond to severe shaking. Moreover, the recordings of strong shaking in buildings allow engineers to understand how the earthquake damage progresses. Also, the assessment of earthquake-resistant design techniques enhances seismic performance prediction, which can be used in upgrading building codes' seismic safety requirements and enabling well-informed safety decisions for building re-occupancy [2]. Consequently, an effective health monitoring technique is needed to assess the condition and damage of these structures during their service life so that economic and human life loss can be avoided.

With the idea about earthquake after-effects, one of our government agencies, Department of Public Works and Highways (DPWH), whose main objective is to ensure the safety of public and private infrastructures, imposed a memorandum circular no. 1 series of 2015 which intends to inform infrastructure owners of the guidelines and implementing rules on Earthquake Recording Instrument (ERI) for buildings. This memorandum is in the provision of section 102 of the National Building Code of the Philippines, otherwise known as PD 1096. Under the principles of sound environmental management and control, this presidential decree declares the state's policy to protect life, health, property, and public welfare. To this end, this code aims to give all buildings and structures a set of minimum standards and requirements to regulate and control their location, size, design, quality of materials, construction, use, occupancy, and maintenance similar to [3]. Consequently, based on the memorandum, the National Structural Code of the Philippines (NSCP 2010) states that every building in seismic zone 4 that is more than fifty 50 m high must have at least three approved recording accelerographs unless the building official waives this requirement and comparable with the study of [4]. It also says that the Philippines must have its earthquake baseline data for validating the seismic design parameters used during any future structural design [5]. These baseline data will be used to improve the awareness of potential damage and behavior of the building during earthquakes and for future earthquakes. In accordance also to the memorandum, the device/instrument for building vibration monitoring is required to immediately transmit data if there will be an occurrence of a magnitude 6 earthquake or greater. The placement and number of sensors are also discussed in the guidelines. A sensor node is needed when the building has a height of less than 50 m, on the other hand, three sensor nodes must be placed each on the top floor, middle floor, and top floor of a building greater than 50 m.

Attuned to the objectives of the DPWH in monitoring the conditions of buildings especially those that are government-owned, the development of a wireless sensor node for earthquake monitoring was made. The wireless sensor node has a sensor that measures vibration and seismic observation on buildings. It also has a microcontroller that gathers, processes, and sends data to the database. The data that the node gathers undergoes a Fast Fourier Transform (FFT) to convert time domain data into frequency domain data. After the conversion, the raw and converted data will then be sent and stored on a cloud database. After the data is stored in the database a website is built to fetch the data stored and present the data in tabular and graphical form.

2. Materials and methods

2.1. Wireless sensor node architecture

In developing a wireless sensor node, some parameters are needed to be considered. All these parameters are listed on the DPWH specification and guidelines on earthquake monitoring instruments. The architecture of the wireless sensor node is shown in Figure 1. The sensor node developed is composed of three parts, the sensor, the power supply unit, and the microcontroller. The sensor gets the vibration data on the environment by placing it in contact with the flooring of a building. Next is the microcontroller, the brain and heart of the system, which receives the data that the sensor reads. If it passes a given threshold, the data is then pre-processed by the microcontroller. Lastly, it will be transmitted to the database through a wireless network for the storing of data. The power supply unit is a key component of the wireless sensor node as it is the one that supplies the power to the microcontroller which also powers up the sensor. Another part of the node that is detached from the actual device is the database which is discussed at the whole system architecture.

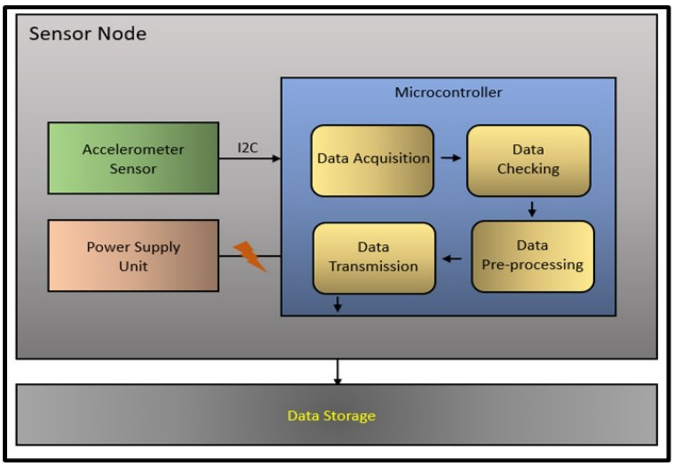


Figure 1. Wireless sensor node architecture.

2.2. System architecture

In the whole system architecture (Figure 2), the research showed how the data flowed on the system from the data on the surroundings to the graphical and tabular representation of data. First, the data gathering, data checking, and data processing happened at the wireless sensor node. Two nodes were placed in a five-story building, one of the nodes read data from the ground floor of the building, and the other one was placed at the top floor. The data that the node gathered, checked, and processed was transmitted to a cloud database through a wireless network. The wireless network used as a medium of transmission is Wireless Fidelity, commonly known as Wi-Fi. The data transmitted were stored in cloud storage. The raw data and the preprocessed data were stored in different storage. For the viewing of data, a simple website was made where the client was able to view the specific data in a tabular or a graphical presentation.

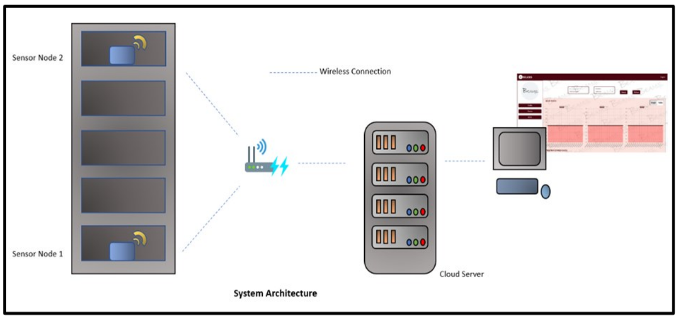


Figure 2. System architecture.

2.3. Components parameters and specification

Once the problem was identified, the next step involved developing a solution and determining the necessary parameters and specifications. In this case, the proposed solution was a wireless sensor node for measuring building vibration data, necessitating a comprehensive study of wireless sensor nodes, building structures, and seismic activity. Key considerations included selecting appropriate sensors and microcontrollers, defining sensor parameters, and specifying microcontroller requirements. Additionally, the sensor node adhered to DPWH specifications and guidelines for earthquake instrumentation in buildings.

2.4. Design and development

Material selection was the next important part of determining the required specifications of the sensor node.

The selection of components was broken down into different sections. The accelerometer, which is the main sensor, the power supply to be used, the microcontroller, which is the brain of the system, the data storage to be used, and the software development for hosting the website. All of these factors were considered during the design and were discussed in the results.

2.5. Hardware testing

The testing of the whole hardware which consisted of the sensor, microcontroller, and the power source was conducted while individual parts were tested for its functionality. It helped for easier troubleshooting if problems would be encountered. The three components were then combined to check their functionality. The sensors were the ones gathering the data dictated by the microcontroller. The microcontroller is the brain of the whole system, which is driven by the power supply unit. The parameters checked were the measuring capability of the nodes and the transmission capability. Also, ensuring that the transmitted data were received properly was tested. Troubleshooting included checking the connection of each component.

2.6. Final testing of sensor node

The integration of the software and hardware was done and tested. This was to ensure that no problems would be encountered. The sensing of the sensor was checked first then the filter capability of the measurement data was checked. The next part tested was the conversion from the time domain to the frequency domain and its transmission to the cloud server for the storage of the data.

3. Results and discussion

3.1. Conduct an inventory of infrastructure facilities with the State Universities and Colleges (SUCs)

Surveying the number of the infrastructure was the first topic of the meeting with the partner SUCs. The application of sensor modes was implemented on the said infrastructures. According to the DPWH memorandum, the number of sensor nodes installed in a certain building was based on how many floors it has. It will be the basis of the number of nodes to be developed along with the partner SUCs.

According to the meeting with the partner SUCs, their buildings have a maximum height of four floors only. As agreed, the Science, Technology, Engineering, and Environmental Research (STEER) Hub building, having six floors, was used for testing the sensor node while the partner SUCs were required to make at least one sensor node.

3.2. Design and development of sensor nodes

Since the research was attuned to the DPWH memorandum, the fabricated sensor node followed the DPWH specification and guidelines about earthquake instrumentation on buildings. Table 1 shows the parameters to follow in building the wireless sensor node and the specifications given by the DPWH.

Table 1. Parameters and specifications of the wireless sensor node.

Data Acquisition Parameters		Miscellaneous Parameters	
Sensitivity	2g	Connectivity	Wireless
Analog to Digital Converter	16 bits	Power	Battery maintained by charger
Axis	3 axis	Storage	Memory card
Sampling Rate	100 samples per second	Environment	IP 67
Recording	Continuous reading of data		

For the data acquisition, the parameters to be considered were the sensitivity of the sensor which determined how sensitive the sensors were in detecting small movements or vibrations on the building, the size of the AD converter which determined the accuracy of the data acquired by the sensor, the number of axes the sensor can read, the sampling rate or the number of samples the sensor and microcontroller read every second and lastly, the frequency of the node to read data of its surrounding. Other parameters to be considered were the connectivity of the node in getting and viewing the data, the power supply of the node, and the environment where the node was implemented. For the specification, it followed the DPWH guidelines which are, a sensitivity of 2 g where g is the gravitational force, an analog and digital converter with 16-bit resolution, can read 3 axes, the x, y, and z-axis, a sampling rate of 100 samples per seconds and can continuously read data. For the other parameters such as connectivity, the DPWH guidelines suggested that it can communicate wirelessly, for power it must be battery powered device that is maintained by a charger, has storage for storing data, and lastly, it must be rated as IP 67.

3.3. Components and devices

3.3.1. Sensor

One of the most important parts of a wireless sensor node was the sensor to be used. Many sensors can be used in detecting and recording earthquake data while some of these sensors that were considered were the velocity sensor, acceleration sensor, seismic, and vibration sensor. Velocity sensors are high-frequency geophones that can record low-frequency signals that shake around 5 Hz or faster. They are often buried under the ground to record vibration. For the seismic and vibration sensors, there are high-grade sensors mainly applied to those big fuel plants and factory equipment. One of the disadvantages of seismic and vibration sensors is its price. Lastly, one of the widely used sensors in detecting vibration is the accelerometer sensor. This sensor can be seen on our mobile devices, the accelerometer is typically available with a (+/-) 2 g maximum range it is also sensitive to low amplitude low-frequency signals. One good thing about accelerometer sensors is that they are usually made from low-cost electronic chips but are good for their seismic applications. Given the sensor that can be used in measuring vibration on a building, an accelerometer sensor was selected to be used in the project. Some of the advantages of the accelerometer against the two other sensors mentioned above were the number of available sensors locally, its cost, and the number of support that can be received in interfacing the sensor to the microcontroller.

3.3.2. Accelerometer sensor

After choosing the accelerometer sensor to be used in the project, the different accelerometer sensors available in the local market were listed according to specific criteria. The sensitivity of the sensor, the analog-to-digital converter, the number of axes it can read, the way of communication with the microprocessor, and the capacity to sample 100 data per second were considered. Table 2 shows the list of accelerometers considered in making the project.

Table 2. Comparison of different accelerometers.

Accelerometers	Requirement Criteria				
	Sensitivity	Bandwidth	AD Converter	Axis	Communication Protocol
Minimum Requirement	*2g*	*0 -100 Hz*	*16 bits*	*3 axis*	*SPI/ I2C*
MMA8451	2g	0 - 400 Hz	14 bits	3 axis	I2C
ADXL 345	2g	0 - 3.2 Hz	13 bis	3 axis	SPI/ I2C
MPU 9250	2g	0 - 100 Hz	16 bits	3 axis	I2C
ADXL 335	3g	0 - 1.6 kHz	16 bits	3 axis	SPI/ I2C
ADXL 357	10g	0 - 1.5 kHz	20 bits	3 axis	SPI/ I2C
ADXL 1003	200g	0 - 6.2 Hz	None	3 axis	SPI/ I2C

After listing different available accelerometer sensors, the minimum requirement for the sensor needs according to the specification given by the DPWH was set and the applicable sensor to be used in the project was selected. The first criterion considered was the number of axes the sensor read, and since all the sensors listed passed the minimum requirement the next thing considered was the sensitivity of the sensor, and for the sensitivity only three sensors passed the given minimum requirement. Given that three sensors passed the requirement for the sensitivity, the analog-to-digital converter was checked wherein only one sensor passed the given minimum requirement and that is the MPU 9250. Given the specifications listed in Table 2, it was then decided to use MPU 9250 since it satisfied the given minimum requirement.

3.3.3. Microcontroller

Another important part of a wireless sensor node is the microcontroller. It stores and executes the algorithm of the wireless sensor node. Everything that the wireless sensor node does is centered at the microcontroller whether it is reading data using the sensor, processing data, or sending data to the main database. One of the important parameters that need to be considered in choosing the microcontroller is the type of connectivity it has. Since the device that needs to be achieved must have wireless connectivity, the focus was on searching for a microcontroller with a built-in Wi-Fi module. The listed microcontroller then was compared based on its processing speed, size of the memory, and capability to continuously read data over a long period. The processing speed and RAM were directly connected to the capability of the node to sample data according to the sampling rate. The size of the ROM was connected to the capability of the node to store data when needed, and lastly, the capability of the node to continuously sample data over a long period.

After listing and comparing all the microcontrollers that can be used in the project shown in Table 3, the microcontroller with the highest specification in the given requirement criteria was chosen. Raspberry Pi 4 is the microcontroller that has the fastest processing speed and bigger variations of RAM size. Raspberry Pi 4 also has the biggest applicable memory storage and is capable of recording data continuously over a long time.

Table 3. Comparison of different microcontrollers.

Microcontroller	Requirement Criteria			
	Processing Speed	RAM	ROM	Continuous Data Reading
Minimum Requirement	*2g*	*0 -100 Hz*	*16 bits*	*3 axis*
MMA8451	2g	0 - 400 Hz	14 bits	3 axis
ADXL 345	2g	0 - 3.2 Hz	13 bis	3 axis
MPU 9250	2g	0 - 100 Hz	16 bits	3 axis
ADXL 335	3g	0 - 1.6 kHz	16 bits	3 axis
ADXL 357	10g	0 - 1.5 kHz	20 bits	3 axis
ADXL 1003	200g	0 - 6.2 Hz	None	3 axis

3.3.4. Data storage

For the data storage, the option was between building their physical storage or paying for a subscription to cloud storage. In choosing, the advantages and disadvantages of using a physical server or using cloud storage in storing the data gathered by the node were weighed. Some of the advantages of using a physical server were the ease of getting the data when needed and having control over the specification of the server such as the size of the RAM and ROM and the speed of the processor. On the other hand, having a physical server has many downsides, and some of these are the maintenance cost since the server must be working most of the time, the power it consumes is high, the requirement for a big facility space, and lastly to develop a design for its security. For cloud storage like the physical server, it can also easily access the data stored by the node, but unlike the physical server, cloud storage is managed not by the owner instead it is managed by the storage provider wherein all the needed software in storing data is preinstalled. The security of the storage is also laid out by the service provider thus, it was decided to use cloud storage for storing data.

3.3.5. Website hosting

One of the things considered was to have a website for the project wherein the data gathered by the node was viewed and presented in a graphical form. Website hosting was one of the important things to consider when deploying a website on the internet. There were two options for deploying a website on the internet. One of them is using a Virtual Private Server (VPS) and the other is using a shared hosting server. To choose the better option, the difference between the two using a specific criterion was made. The criteria used to compare was the security it can provide once the website is deployed. This is to ensure that all the data that the website is going to show cannot be altered without the permission of the administrator. Next, was the size of RAM it can provide to ensure the smooth operation of the website when it is deployed and visited by other people to gather data. Another criterion was the cost of the server and the difficulty of maintaining such a type of server. Eventually, the use of the shared hosting server was selected. One of the main reasons for choosing the shared hosting was the way the server is maintained. The service provider maintains a shared hosting server, so the user does not need to worry about maintaining a server. Shared hosting also has a pre-installed application to be used in deploying the website, unlike the VPS in which the owner must first install all the programs needed for the project.

3.3.6. Sensor node power supply

One of the important things that needs to be considered in building a wireless sensor node is the power supply since the node must read data every time a switching power supply is needed. A switching power supply is a type of power supply that can be powered in two ways, one is through the local outlet and a battery pack. Since during earthquakes, there is a bigger chance that there is a power interruption the battery pack would be able to power up the device and still record the data. One of the characteristics of an earthquake is that it will only happen for a smaller time frame, thus an uninterruptible switching power supply is needed. The switching between the local outlet and the battery pack should have no time lag which means the system will power off for only a split second to avoid important data loss. The current rating of the power supply also needs to be considered in this project. A power supply must have a current rating of 3 A above for it to supply enough current to the Raspberry Pi to power up the MPU-9250 accelerometer sensor, and a fan.

Figure 3 shows the power consumption of each device used in the node. The power supply can be powered by our local outlet or a battery pack, the input of the power supply must be 5 V and an input current of 2 A. Then, the power supply must be able to supply 5 V and a minimum of 3 A of power or 15 W of power to power up the Raspberry Pi. Lastly, the Raspberry Pi would power up the fan and the sensor through its pin, however, it does not mean that the sensor and the fan have the same power rating. The voltage rating of the sensor is 3.3 V which can be supplied by pin 1 of the Raspberry Pi and it must have an input current of 450 uA which can be compensated by pin 1 with an output current of 50 mA. On the other hand, the fan is rated by 5 V and a maximum of 0.2 A of current. The fan is then connected to pin 2 of the Raspberry Pi, pin 2 can produce 5 V of voltage and since it is connected to the main power supply of the Raspberry Pi it can supply 3 A of current.

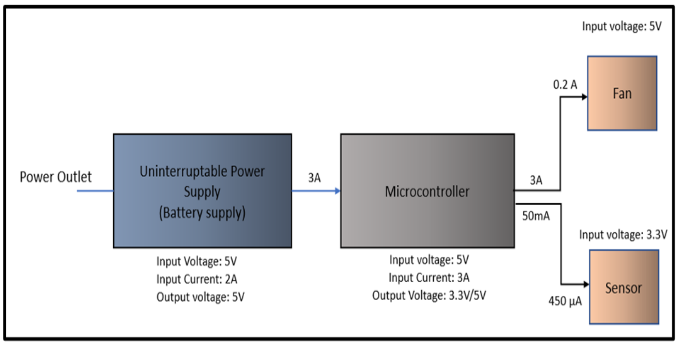
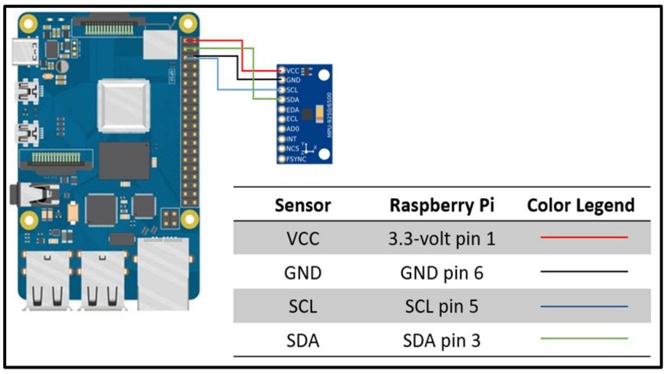


Figure 3. Power consumption diagram.

3.3.7. Interfacing of sensor on the microcontroller

For the interfacing of the MPU-9250 accelerometer sensor to the Raspberry Pi, the communication protocol to be used should be considered. Two available communication protocols were used to establish a short-distance synchronous communication between two devices, those two are the Inter-Integrated Circuit widely known as I2C, and the SPI short for Serial Peripheral Interface. In choosing the communication protocol, the I2C was chosen because of its simplicity. The I2C protocol uses only two wires for the transfer of data, which lessens the problem of lost connection. It is also the only available communication protocol that can be used by the MPU-9250 accelerometer sensor.

To interface the MPU-9250 with the Raspberry Pi, it is important to know the pins that are needed for the I2C communication protocol. MPU-9250 accelerometer sensor module has ten pins in total and only four of those pins were used. The four important pins are the VCC, GND, SDA, and the SCL. VCC and GND are the pins to power up the device. SDA or the serial data line was the one where the data was passed from the sensor to the microcontroller, connected to the SDA or pin 3 of the Raspberry Pi. Lastly, SCL, or the serial clock was the one responsible for the timing of the Raspberry Pi and the sensor in transmitting data, which made sure that the data was read at a correct reading or sampling time. Figure 4 shows the connection between the Raspberry Pi and the MPU-9250 accelerometer sensor module.



Sensor	Raspberry Pi	Color Legend
VCC	3.3-volt pin 1	red
GND	GND pin 6	black
SCL	SCL pin 5	blue
SDA	SDA pin 3	green

Figure 4. The physical connection between the microcontroller and accelerometer sensor.

3.3.8. Software/program development

Connecting our Raspberry Pi, sensor and power source was only the first step in developing the whole sensor node. A program or a set of commands is a must for our sensor to gather data and for our Raspberry Pi to process and save data on a storage device. A set of commands was programmed to the Raspberry Pi and carried out the following command.

Three major codes needed to be developed, the code for gathering data, the code for processing data, and lastly, the code for sending data to our database. Other than those three major codes, some minor codes were needed for the node to carry out its application. Some of those codes are the calibration of the sensor and the connection of the node to the cloud storage. The following steps will be discussed individually.

3.3.9. Connection of the node to the cloud storage

For a faster and better construction of the program for the node, it was decided to make a function for each step that would happen inside the node. The first step was the connection of the node to the cloud storage. For the connection of the node to the cloud storage, a library that can be used to wirelessly connect the node to the cloud storage was required. The PyMySQL library was used which is available for the Python programming language. PyMySQL is a Python library package that contains a pure-Python MySQL client library that is based on PEP 249. This library will enable the node to remotely connect to a MySQL database over the internet. In connecting the node to the database, the hostname or IP address of the MySQL database, the username, and password used to connect to the database, and lastly, the name of the database wherein the data will be stored must be specified. After all the needed things for database connection were identified, the PyMySQL would be tested to connect to the database. If the connection is successful, the code would then return the connection and cursor variable then proceed to the next part of the code. On the other hand, if there is an error in the connection the node will then retry to connect to the database.

3.3.10. Data gathering

For the data gathering, the node would first initialize the library used in the gathering of data using the MPU-9250 accelerometer sensor, this library is called mpu9250-jmdev. This library connected the MPU sensor to the Raspberry Pi using the I2C communication protocol and has a built-in sensor calibration as well. After initializing the library to be used, the required settings are the I2C address of the sensor and the sensor sensitivity were specified. The sensor I2C address was checked using the Raspberry Pi terminal. Typing the syntax “sudo i2cdetect -y 1” showed the address of all the I2C devices that are connected to the Raspberry Pi. After specifying all the needed settings for the sensor, the accelerometer sensor was calibrated. In sensor calibration, the sensor read data to the surroundings and processed the data using a calibration algorithm that supplied a certain offset value used as a reference in the actual reading of data. After calibrating the sensor, the node started to read data, and every time the sensor read data, the node was also recording the date and time of data reading. All the x-axis, y-axis, z-axis, and date-time data were stored on an array. The node checked the number of data recorded on the array and if there are a hundred data in the array all the data were then checked whether one of the values recorded reached or passed a certain threshold. Lastly, the function for reading data returned data based on whether the values reached a certain threshold. If the data reached or passed the threshold, the data returned would be a Boolean “True” following the data of the date-time, x, y, and z-axis. On the other hand, if the data was not able to pass or reach the given threshold, the data returned was the same data except that the returned Boolean will be ”False” which means that the data would not be sent to the database.

3.3.11. Data processing

In the data processing of the wireless sensor node, the main purpose was to change the time domain data to frequency domain data. To do this, FFT was performed on the raw data gathered by the sensor node. In the development of the FFT algorithm, different libraries were used. Those libraries are SciPy and NumPy where the former is a Python library that can provide algorithms for optimization, integration, interpolation, eigenvalue problems, algebraic equations, differential equations, statistics, and many other classes of problems. On the other hand, the NumPy library offers comprehensive mathematical functions, random number generators, linear algebra routines, and Fourier transforms. The statistical library for Python to get the mean of the group of data was used. Getting the mean of the raw data and subtracting it from the same data removed the DC offset of the resulting frequency data. After removing the DC offset of the raw data, the data proceeded to the FFT algorithm and changed the time domain data to frequency domain data. After the data transformation, the data was stored in an array and then returned to the next process which was the sending of data to the database.

3.3.12. Data sending

The last part of the program was for the code for the sending of data to the database. At the start of the program, the node was already connected to the cloud storage wherein it gathered data to be saved. The raw data came from the returned data of the data gathering and the process data came from the returned data of the data processing. These two data were sent to the cloud storage. In sending the data to the cloud storage, the node used the PyMySQL libraries, and the name of the table and the column were specified wherein the data must be placed. Given the following, the node was ready to send the data to the database. If there is an error in the sending of data, the data will be stored in the node for recording.

3.3.13. Main program function

The main function was the main loop of the program where exchanging of data between different other functions happens. It is also where the node decides whether the data will be sent to the database or not. The main function was the code that runs other functions to do their respective jobs, and also where the whole process happens. The first function was called the connection of the database to the cloud storage. Since connecting to the database sometimes takes a lot of time, it was decided to connect only one time once the system was powered up and simply ping the database to avoid disconnection. After connecting to the database, the sensor started its data acquisition wherein it ran in a loop until there were a hundred data read. The data were checked whether the data acquired passed the given threshold. If the data did not pass the given threshold, the program would not proceed in processing or sending the data. On the other hand, when the data passed the given threshold, the program proceeded to the data processing then the raw data and the process data were sent to the database and the whole process looped again starting from the part of data acquisition. The program would not end until the researcher stops the program.

3.3.14. Identifying the threshold of the sensor

The last part of the development of the wireless sensor node was to identify what is the accelerometer threshold wherein the gathered data of the sensor would be sent and saved on the database. This process was added to the design to solve the problem of database space exhaustion. Designing the node in a way where the data would be sent and saved on the database once the data gathered passed the threshold set. To get the threshold to be set on the node, the data on the building wherein the node was to be deployed was initially gathered. This initial gathering of data determined the stable reading of the vibration or movement of the building. For the initial data gathering, the node was deployed at the ground and top floor of the STEER Hub building and gathered data continuously for some time. Figures 5, 6, and 7 show the initial data-gathering setup for the identification of the threshold of the sensor is shown

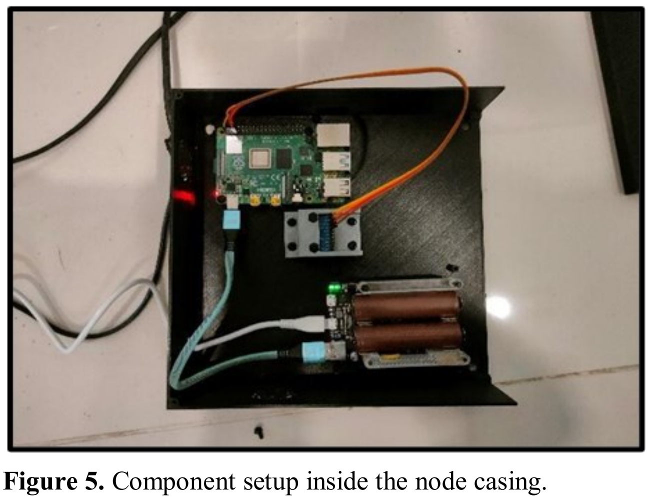


Figure 5. Component setup inside the node casing.

Figure 6. Wireless sensor node setup at the top floor of the STEER Hub building

Figure 7. Wireless sensor node setup on the ground floor of the STEER Hub building

After gathering data on the top and the ground floor of the STEER Hub building, the data gathered by the node on the database was downloaded. Using Python, the maximum and minimum values of the x, y, and z-axis were determined and set as a threshold of the wireless sensor node. The data gathered at the top floor of the building was processed separately from the data gathered at the ground floor of the STEER Hub building.

For the process of getting the maximum and minimum value gathered by the node, the JSON library was imported. This library is used to open and fetch data on a JSON file. After importing the needed library, the downloaded JSON file contains the data gathered and this data is saved separately at different arrays. Lastly, the array that contains each gathered data for the x, y, and z-axis of the ground floor, and the x, y, and z-axis of the top floor, the maximum and minimum values gathered in each axis were identified. The result of the Python script was then printed at the terminal. Figure 8 shows the result of the Python script. The result shows the maximum and minimum value of the data gathered in each axis on the ground and the top floor of the STEER Hub building.

Figure 8. Maximum and minimum value of the data gathered on each floor.

3.4. Development of a central station for all nodes

3.4.1. Website development

For the presentation of data gathered by the node, it was decided to make a website that showed the data in a tabular or a graphical presentation. In developing a website, three things needed to be considered which were the frontend, the backend, and the database to be used. For the database of the website, the MySQL database was used since the node uses MySQL in storing data. The node storage and the website storage should be the same since the data to be shown on the website was the one the node gathered. Next was the frontend, which was the part of the website wherein the client interacted. It served as the graphic interface of the website which also showed the tabular and graphical list of data gathered by the node. The third part of the website development is the backend. The backend is the part of the website that talks with the frontend and the database, it serves as the communication channel between the frontend and the database.

3.4.2. Frontend development

In developing the frontend of the website, it is necessary to choose what framework would be used. Many familiar frontend frameworks can be used, some of those frameworks are React, Angular, Vue, jQuery, and many others. The frameworks listed have their pros and cons and it depended on familiarity with the frameworks used in developing the frontend of the website. Using angular frameworks in making the program, familiarity with typescript language was needed. Typescript language is a language that is almost like JavaScript which is used in programming the logic of the website page. Familiarity with the Hypertext Markup Language (HTML) was also needed which is used to structure web page content and the Cascading Style Sheets (CSS) used to style web page content. The main function of the frontend was to interact with the clients of the websites. When a client needs something, the client will relay its request on the graphical interface of the frontend then the frontend will pass the request to the backend. In developing the frontend, it was ensured that the web page was working properly. Some of the factors that needed to be working were the button for navigating to another part of the website, and the button for the login and the register must perform its job correctly. One of the most difficult parts in developing the frontend was the part wherein it will interact with the backend, wherein the front end must know the type of data the backend sends for it to be delivered to the client.

3.4.3. Backend development

For the development of the backend for the website, it was necessary to choose the framework to be used. Several backend frameworks can be used for the project, which includes Django, Flask, ASP.Net, Ruby on Rails, and many more. Considering the familiarity with the backend framework, Django was favored because it used the same Python language, and has some advanced features that can be used in development. One of the important things that were considered in making the backend of the website was the security of the website. This is to ensure that no one can delete or destroy the data stored in the database by the node. The backend served as the connection between the frontend and the database. The request of the client would be transmitted from the frontend to the backend, which would then retrieve the client's required data from the database and return it to the frontend.

3.4.4. Interaction between the frontend, backend and database

One of the important things in making a website was to build the interconnection between the three parts of the website: the frontend, the backend, and the data storage. For a website to work perfectly, the interconnection between these three parts must be synchronized perfectly. The process of accessing the data read by the node was as follows. When the client wanted to look for the data read by the node, the client first logged in to their account if they had an account. If not, the client had to register for an account where they input some important credentials such as username, password, and email. This process all happened at the front end of the system. After all the needed credentials were typed in, the client then clicked the register button. The client’s input in the form at the front was sent to the backend where the data was checked before it was permanently saved to the database. If the data was successfully stored in the database, the backend then sent a response to the frontend about the successful creation of a new user. This same process happened with every request that the client made at the frontend. A simple illustration is shown in Figure 9 which shows the interaction between the database, frontend, and backend of the website.

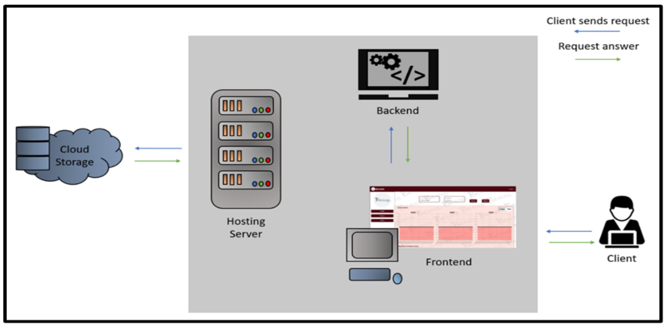


Figure 9. Illustration of how the front end, back end, and database interact with each other.

3.4.5. The website

The website was composed of two major parts: the authentication part and the dashboard part. The authentication part was what the client first saw when they visited the site. It was composed of the home page, login page, and registration page. On the other hand, the dashboard part was what the client saw when they logged on to the site. The dashboard part was composed of the node page, team page, and SUCs page.

3.4.5.1. Authentication page

The authentication part was the first thing the client saw when they planned to visit the site. The first part of the authentication part was the homepage, where a short description of the Building Research and Innovation Development Goals for Engineering State Universities and Colleges Program, which included the project, was shown. The homepage also discussed what the project was all about, and at the bottom part of the page, there was a simple video about the project. In this part of the website, the SUCs that joined and collaborated on the project were shown. At the navigation bar of the homepage was where the button for the login and registration pages was located. The login page was where the client input their email and password to visit the dashboard part of the website. For those new clients who wanted to view the data gathered by the node, they first had to create an account on the registration page. Some credentials such as email, username, and password were needed to successfully create an account. Missing one of these requirements would result in an account creation error. The email and username the client provided had to be unique to create an account. Retyping the password was also needed to check whether the client typed the correct email they decided to use.

3.4.5.2. Dashboard page

The dashboard page was one of the most important parts of the website. The dashboard page was composed of three more parts: the node page, the team page, and the SUCs page. The teams and SUCs page was where the team management was shown, while the SUCs page displayed the seven other SUCs that collaborated on the project. A button linked to the Facebook and official website of each SUC was also included on this page. Lastly, and most importantly, was the node page. In this part, the raw data was presented in a table and graphically. At the top of the node page was a search bar where you could search for the date, and the data read on that date was shown in the table and graph. A refresh button was also included to refresh the view of the page. A button named "processed" was also seen on the node page that displayed the processed data. Figures 10 and 11 show the dashboard page and its parts.

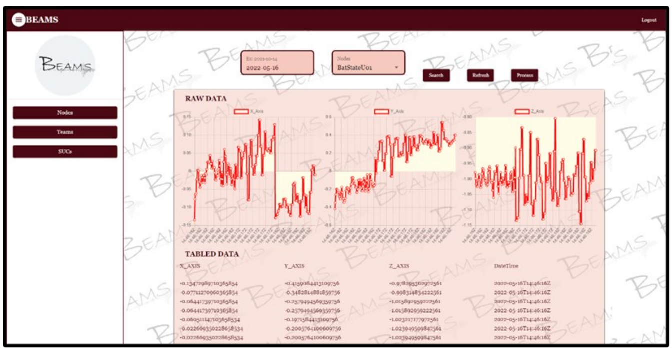


Figure 10. Node page with the raw data.

Figure 11. The page that shows the processed data.

4. Conclusions

The WSN was designed according to the following parameters outlined by the DPWH standard memorandum. These standards were shared and discussed with partner SUCs to ensure proper alignment. The components were chosen carefully to meet these requirements, and improvements were made by reducing and upgrading the microcontroller.

Despite facing challenges with component availability, the WSN was successfully built and gathered the necessary data, although with slightly reduced processing power. This allowed the sensor node to capture essential vibration data from a building and store it for further analysis.

Looking ahead, the researchers recommend using an industrial-grade accelerometer to improve the sensor's accuracy and reliability, even though it comes at a higher cost. This upgrade would enhance the WSN's performance, making it more effective for structural health monitoring in the future.

Acknowledgment

The authors would like to thank the Manufacturing Research Center (LIKHA FabLab) and the Electronic Systems Research Center of Batangas State University The National Engineering University for the support of tools and measurement equipment used for this investigation. Also, the seven SUCs, Bulacan State University, Don Honorio Ventura State University, Kalinga State University, Mountain Province State Polytechnic College, Tarlac State University, University of Rizal System, and University of Southern Mindanao, which are the collaborators of this project.

References

[1] Arya AS, Karanth A, Agarwal A. Hazards, disasters and your community- A primer for parliamentarians [Internet]. New Delhi: National Disaster Management Division, Ministry of Home Affairs, Government of India; 2012 Jan. 62 p. Report. Available from: https://www.undp.org/india/publications/hazards-disasters-and-your-community-primer-parliamentarians

[2] Harrison P, Christensen I, Coffey E, Hochstein J, Rojas Gregorio JI. Earthquake monitoring and response from space: the TREMOR concept. In: Proceedings of the ASTRO-CASI conference 2008; 2008 Apr 29; Montreal, Canada. Ontario, Canada: Canadian Aeronautics and Space Institute; [date unknown]. p. 15.

[3] Çelebi M, Page R. U.S. Geological Survey [Internet]. Monitoring earthquake shaking in federal buildings | USGS fact sheet 2005-3052; 2005 Dec 19. Available from: https://pubs.usgs.gov/fs/2005/3052/.

[4] Paul SM, Jayaguru C. Health monitoring of concrete specimens using smart aggregates. I Manag S J Struct Eng [Internet]. 2016;5(3):25-32. Available from: https://doi.org/10.26634/jste.5.3.8253

[5] Department of Public Works and Highways [Internet]. Guidelines and implementing rules on earthquake recording instrumentation for buildings | department of public works and highways; 2015 Mar 12. Available from: https://www.dpwh.gov.ph/dpwh/references/guidelines_manuals/earthquake_recording