A

Project Report

Entitled

IoT Based Health Monitoring System for Industrial Motors

Submitted to the Department of Electronics Engineering in Partial Fulfilment for the Requirements for the Degree of

Bachelor of Technology

(Electronics and Communication)

: Presented & Submitted By :

ANURAG KHARWAR, AMIT KUMAR, DINESH CHOUDHARY

Roll No. (U19EC061, U19EC073, U19EC078)

B. TECH. IV(EC), 8th Semester

: Guided By :

Dr. ANAND D. DARJI

Associate Professor, DoECE

(Year: 2022-23)

DEPARTMENT OF ELECTRONICS ENGINEERING

SARDAR VALLABHBHAI NATIONAL INSTITUTE OF TECHNOLOGY

Surat-395007, Gujarat, INDIA.

Sardar Vallabhbhai National Institute Of Technology

Surat - 395 007, Gujarat, India

DEPARTMENT OF ELECTRONICS ENGINEERING

CERTIFICATE

This is to certify that the Project Report entitled "**IoT Based Health Monitoring System for Industrial Motors**" is presented & submitted by ANURAG KHARWAR, AMIT KUMAR, DINESH CHOUDHARY, bearing Roll No. U19EC061, U19EC073, U19EC078, of B.Tech. IV, 8^{th} Semester in the partial fulfillment of the requirement for the award of B.Tech. Degree in Electronics & Communication Engineering for academic year 2022-23.

They have successfully and satisfactorily completed their **Project Exam** in all respects. We, certify that the work is comprehensive, complete and fit for evaluation.

Dr. ANAND D. DARJI
Associate Professor& Project Guide

PROJECT EXAMINERS:

Name of Examiners	Signature with Date
1. **Dr. A. D. Darji**	________________
2. **Dr. P. K. Shah**	________________
3. **Dr. Deepak Joshi**	________________
4. **Dr. A. Aacharya**	________________
5. **Dr.Nitin Chatterji**	________________

Dr. (Mrs.) Rasika Dhavse
Head, DoECE, SVNIT

Seal of The Department
(May 2023)

Acknowledgements

I would like to express my profound gratitude and deep regards to my guide Dr. ANAND D. DARJI for his guidance. I am heartily thankful for suggestion and the clarity of the concepts of the topic that helped me a lot for this work. I would also like to thank Prof. Dr. (Mrs.) Rasika Dhavse, Head of the Electronics Engineering Department, SVNIT and all the faculties of DoECE for their co-operation and suggestions. I am very much grateful to all my classmates for their support.

Anurag Kharwar
Amit kumar
Dinesh Choudhary
Sardar Vallabhbhai National Institute of Technology
Surat

May 2023

Abstract

The vibration monitoring is a process where a particular parameter is observed for a period of time which could be an indicative of the machine health. If the period of observation is good enough, the vibration monitoring can provide the signs of a machine failure. Nevertheless, to say, vibration monitoring is an important method to avoid any large after effects of failure, since the failure can be detected well in advance. Early detection of failure also puts forward an important step towards safety, both of the machine and the person handling the machine. The role of vibration monitoring in cost control is also to be looked into closer, since the higher the failure rate is, the higher will be the cost of it. So, by early detection of faults, maintenance cost of the machine can be saved by a large amount.

This project aims to use vibration patterns produced by a machine to predict the health status of the same machine. This is accomplished through IoT integration, which allows for continuous monitoring of machine status. The Internet of Things (IoT) is a network that connects various devices. Smartphones, computers, data, people, processes, and physical items such as gadgets, machinery, and appliances are examples of these things. The Internet of Things (IoT) can connect a variety of manufacturing devices to sense, recognise, process, integrate, function, and network. This highly integrated intelligent cyber-physical environment provides industrial firms with a plethora of opportunities and benefits.

A hardware needs be added to a spinning machine in order to make it intelligent for vibration analysis. It must be a piece of hardware capable of calculating the machine's vibrations and transmitting them to the network for further analysis. This is accomplished by mounting an accelerometer sensor to the equipment under test. The accelerometer measures vibration patterns in all three dimensions- x, y and z. These patterns are then transferred to a micro controller via I2C connection. The microcontroller produces the FFT of the spectrum, which will show as the different frequencies produced by the equipment. The FFT contains harmonics of the fundamental frequency component, which will point us to any possible fault in the equipment. The data from the microcontroller is then transferred to the internet by using a Wi-Fi module. This will display the data in the personal computer. The main concentration here is to produce a cost-effective method to accomplish all the above stated requirements with minimal hardware.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

FFT	Fast Fourier Transform
SPI	Serial Parallel Interface
ADC	Analog Digital Convertor
MCU	Microcontroller Unit
MPC	Microprocessor Unit
I2C	Inter-Integrated Circuit
I/O	Input/Output
IDE	Integrated Development Environment
IR	Infrared
ML	Machine Learning
AI	Artificial Intelligence
RCM	Reliability Centered Maintenance
RMS	Root Mean Square
CPM	Cycle per Minute
DFT	Discrete Fourier Transform
MEMS	Micro Electronic Mechanical System
ISO	Internation Organisation for Standardization
CPU	Central Processing Unit
WLAN	Wide Local Area Network
IoT	Internet of Things

Chapter 1
Introduction

In this chapter, we provide a primer on remote health monitoring system for industrial motors, exploring its inspiration and ultimate goal while also reviewing a selection of the relevant literature.

1.1 Introduction

Vibration diagnostics is a non-destructive approach for monitoring the status of machinery in operation. While in operation, all machines vibrate to some extent, and the majority of them produce undesirable vibrations, which are referred to as incorrect vibration data. Each machine must be maintained if it is to function consistently for the duration of its intended life. Operational health is an important, but frequently overlooked, aspect of the life of any large, expensive machine. While the building of a machine, which includes planning, development, manufacturing, and assembly, normally takes two to three years, the operational life of a heavy machine is estimated to be more than twenty years.Yet, very little care is taken for the predictive maintenance of the same machine. Appropriate maintenance is required for a smooth-running condition of a machine. Predictive maintenance, like the old adage goes, isn't done to repair broken equipment; rather, it's done to prevent it from breaking down in the first place through constant monitoring [1]. A machine is also built to perform efficiently, consistently, and safely.
The maintenance role can be summarised in three points as below:

1. **Achieve maximum productivity**
    - One strategy to attain maximum productivity is to ensure that the machine runs continuously and satisfactorily throughout the duration of its expected lifetime.
    - Another practice is to reduce the down times for maintenance and repairs, thereby achieving a higher machine utilisation.
2. **Optimise machine performance**
    - A machine in slick and well functional condition will produce a higher quality product.
    - An optimised machine will be always an economical one. It will take less price if any fault is detected in the early phase itself.

3. **Ensure operational safety**

    - An important matter to consider here is the safety. Not only the safety of the equipment, but the operator also.
    - A predictive maintenance method not only takes care of the productivity of the machine, but also the safety.

The vibration monitoring is a process where a particular parameter is observed for a period of time which could be an indicative of the machine health. If the period of observation is good enough, the vibration monitoring can provide the signs of a machine failure. Nevertheless, to say, vibration monitoring is an important method to avoid any large after effects of failure, since the failure can be detected well in advance. Early detection of failure also puts forward an important step towards safety, both of the machine and the person handling the machine. The role of vibration monitoring in cost control is also to be looked into closer, since the higher the failure rate is, the higher will be the cost of it. So, by early detection of faults, maintenance cost of the machine can be saved by a large amount [1].

1.2 Maintenance Types

There are basically four types of maintenance employed in an equipment nowadays, as listed below. The distinguishing factors in these four are the time when maintenance is performed, the method used and the cost.

1. **Reactive maintenance**
   Equipment maintenance is critical for long-term, trouble-free operation. Several types of maintenance have been identified as technological progress has progressed, the implementation of which is subject to a variety of scenarios that must be carefully considered. However, when it comes to basic maintenance tasks, expenses should always be weighed against safety. As a result, small and supported equipment is still discarded in a maintenance-free manner – that is, until it fails. Domestic appliances (frequent inspection of a vacuum cleaner or microwave is not done on a regular basis) and tiny (and properly supported) pumps, for example, are examples of this type of maintenance. Reactive maintenance is the term for this form of maintenance [8].

2. **Preventive maintenance**
   The method of periodic maintenance exams or repairs, known as preventative maintenance, has been recognised for more exclusive equipment with more expensive operation. Cars with a service book and specific service activities that

are distance-based or completed in specific periods of time are examples of this form of maintenance. In this way, a large number of bulky industrial facilities are also preserved. The goal is to prevent the machine from failing. The rate of failures of similar equipment is used to calculate the time to repair – the failure rate mimics the so-called mean time to failure. This demonstrates a flaw in this technique because it is difficult to estimate the period between repairs - many devices fail before they are intentionally fixed, and some are reviewed or repaired ineffectively, increasing the expense of maintenance [8].

3. **Predictive maintenance**
   A machine is fixed when it needs to be fixed, not at predetermined intervals. Of course, it is necessary to be aware of this condition and, as a result, to see the machine in action, i.e., to perform monitoring and diagnostics. We can avoid unintentional closures and failures by using this strategy. The important clue is having the right information at the right moment. If we know which part of the machine has to be replaced or repaired, we can obtain extra parts, assemble it for personnel, and complete the project on time. A planned shutdown is both faster and less expensive than one caused by equipment failure or even an accident. Other benefits of predictive maintenance include enhanced equipment lifetime, increased safety, fewer accidents with negative environmental consequences, improved spare part administration, and so on [8].

4. **Proactive maintenance**
   In addition to the previous form of care, this one entails addressing the significant condition's root causes. The fundamental goal of remedial actions is to identify and address the underlying cause of the defect, rather than existing signs of the fault (e.g., a damaged bearing) (e.g., damage of the bearing has got up due to a misalignment of the machine).

Other recent maintenance systems include RCM (Reliability Centered Maintenance), which is popular in aviation, and others . When predictive and proactive maintenance is in place, it's critical to keep track of the machine's current state. The five stages of the maintenance process are as follows:

1. **Defining the early condition :** When the machine is in good working order, a thorough measurement is performed, providing the basic reference values for further comparison.

2. **Continuous Monitoring :** On the machine, there are well-defined places where vibrations should be measured at regular intervals. The entire value of vibration is usually determined. Without analytic expertise, a competent worker can complete this task.

3. **Detection :** The information gathered through continuous monitoring is easily quantified. Alarm limits are specified for each value obtained. When the planned alarm limit is exceeded, a problem is detected.

4. **Analysis :** Following the detection of the problem, extensive measurements and research (drift assessment, FFT analysis, phase analysis, and so on) are carried out, allowing for a clear picture of the problem and its root cause.

5. **Recommendation :** Once the root cause of the problem has been identified, cost-effective corrective schedules can be advised and implemented.

1.3 Basics of Vibration

In mechanical systems, vibrations can be caused by forces acting on the system, such as external forces or internal forces due to imbalances or misalignments. These vibrations can cause wear and tear on the system, and in some cases, can lead to catastrophic failure [9].

1.3.1 Definition

Vibration or oscillation refers to any motion that repeats itself over a period of time. The motion of a tugged string and the wavering of a pendulum are both examples of vibration. The study of oscillatory motions of bodies and the forces that cause them is central to vibration theory. In very modest terms vibration is an oscillatory motion of minor amplitude. Every body has a vibration signal that is linked to each of its qualities. As a result, each machine generates its unique vibration signal, which contains data from each of its components. In other words, a machine's vibration signal is the sum of the vibrations of its individual components. These are the simple ones, and they show pure oscillations. Three measures can be used to describe vibration patterns:

1. **Amplitude**
   The amplitude of a vibration is a measure of how far a mass can move from its neutral state. It is the vibration's intensity, and it reveals the vibration's severity. Wave amplitude can be measured in a variety of methods. Because the curve is symmetric, there is also an amplitude -A mm in the opposite direction if the vibration pattern has a peak amplitude of A mm. As a result, the curve has a peak-to-peak displacement of 2A, which corresponds to A mm up and A mm down. The RMS (root-mean-square) value, which is a bit more composite, is the third way of relating the amplitude. The square root of the average of the squares of the waveform values is what it's called. The RMS value for a sine wave is

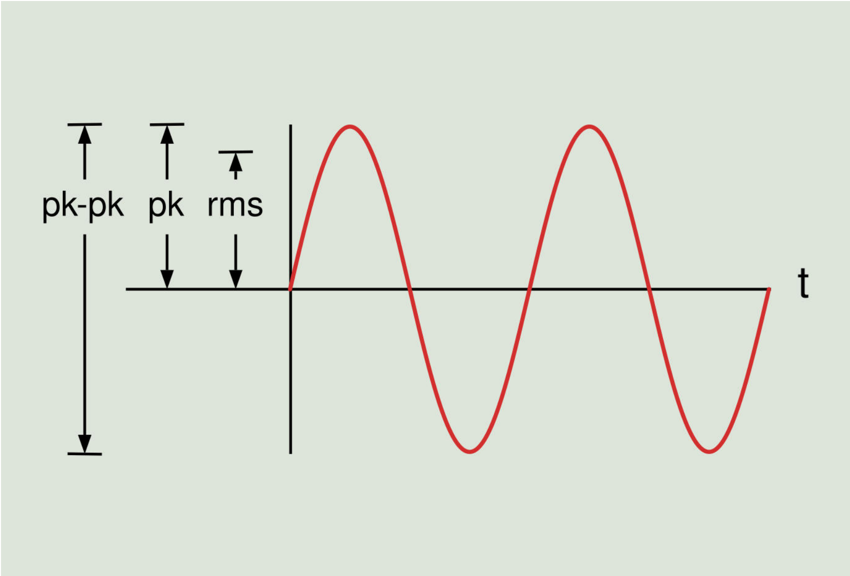

Figure 1.1: Characterises the amplitude of vibration [1]

equal to [9] . 0.707 of the peak value, although this is only true for sine waves. The RMS value is used to quantify the waveform's energy. The characterises the amplitude of vibration shown in "Figure 1.1" [1].

2. **Frequency**
   The length of time it takes for the vibration pattern to repeat is always the same. The oscillation period (typically measured in seconds or milliseconds) indicates that the machine has completed a rotating cycle. The frequency (F = 1 / P) is the inverse of the period and is usually expressed in Hz (cycles per second) or CPM (cycles per minute) (cycles per minute).

3. **Phase**
   It's a measure of the difference in time between two sine waves. Despite the fact that the phase represents a difference in time, it is always expressed in terms of angle, degrees, or radians. This is a method of time standardisation in which a full wave cycle is used as a reference without regard for its true period of time. Phase offset or phase shift refers to the difference in phase between two waves. A 360-degree phase shift is defined as a one-cycle delay or a full wave period of phase shift. The phase shift might be positive or negative, indicating that one waveform is behind or ahead of another. These are referred to as phase lag and phase lead, respectively. The characterises the phase of vibration shown in "Figure 1.2" [1].

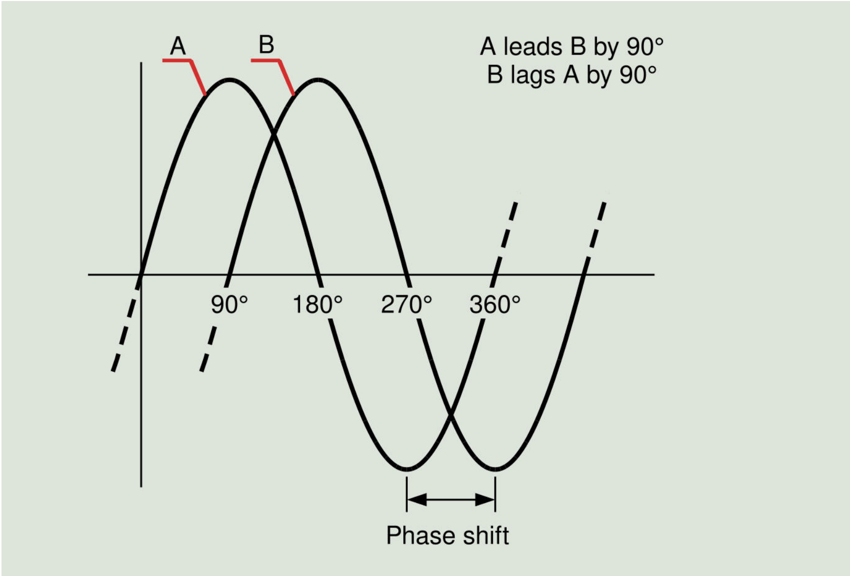

Figure 1.2: Characterises the amplitude of vibration [1]

1.3.2 Vibration Pattern Classification

Vibration can be classified into the following catergories:

1. **Free and Forced Vibration**
   The subsequent vibration is recognised as free vibration when a system is left to vibrate on its own after an initial disruption. There are no external forces acting on the system. Forced vibration is the oscillation that occurs in equipment such as diesel engines. A situation known as resonance happens when the frequency of the outside force equals one of the system's intrinsic frequencies, and the system suffers dangerously enormous oscillations. Building collapses, bridge collapses, turbine failures, and aviation wing failures have all been linked to the occurrence of resonance [9].

2. **Undamped and Damped Vibration**
   If no energy is squandered or dissipated in friction or other resistance during oscillation, the vibration is known as undamped vibration. However, if any energy is lost in this way, the vibration is said to be damped. Many physical systems have so little damping that it's easy to overlook it for most engineering applications. [9]

3. **Linear and Nonlinear Vibration**
   The resultant vibration is known as linear vibration if all of the primary components of a vibratory system, the mass, the spring, and the damper, act linearly. Nonlinear vibration occurs when one or more of the basic components behaves non-linearly. The linear and nonlinear differential equations that govern the be-

haviour of linear and nonlinear vibratory systems are, respectively, linear and nonlinear. The concept of superposition holds true if the vibration is linear, and mathematical methods of analysis are well-known. Because nonlinear vibrations are not subject to the superposition principle, analysis tools are limited. Because all vibratory systems tend to act non-linearly with cumulative amplitude of oscillation, when dealing with real-world vibratory systems, a familiarity with nonlinear vibration is desirable [9].

4. **Deterministic and Random Vibration**
   The excitation (force or motion) acting on a vibratory system is said to be deterministic if the value or amplitude of the excitation (force or motion) is known at any given time. The vibration that follows is referred to as deterministic vibration. In some circumstances, the excitation is non-deterministic or random, meaning that the value of the excitation at any given time cannot be predicted. In these circumstances, a large number of excitation data may show some statistical regularity [9].

1.4 FFT

FFT stands for Fast Fourier Transform, which is a widely used algorithm for efficiently computing the discrete Fourier transform (DFT) of a sequence of data. The DFT is a mathematical technique that allows us to analyze the frequency components of a time-domain signal. The FFT algorithm was invented by Cooley and Tukey in 1965, and it has since become a cornerstone of digital signal processing [2].

Here are the basic steps involved in the FFT algorithm:

1. **Prepare the input data:** The input data is typically a sequence of N complex numbers (or real numbers, if the input signal is real). The number N must be a power of 2 for the FFT algorithm to work efficiently.

2. **Divide and conquer:** The FFT algorithm is a divide-and-conquer algorithm that recursively divides the input data into smaller and smaller subproblems. The algorithm works by computing the DFT of the even-indexed elements and the odd-indexed elements separately, and then combining them using complex exponential factors.

3. **Combine the results:** After computing the DFT of the even and odd subproblems, the algorithm combines them using complex exponential factors to compute the final DFT of the entire sequence.

4. **Interpret the results:** The output of the FFT algorithm is a sequence of N complex numbers (or real numbers, if the input signal is real), which represents the frequency components of the input signal. The magnitude of each complex number represents the amplitude of a particular frequency component, while the phase represents the relative phase of that component.

The FFT algorithm has many applications in signal processing, such as filtering, spectral analysis, and image processing. It is also widely used in numerical analysis, scientific computing, and other areas of mathematics and computer science.FFT of two frequency signals shown in "Figure 1.3" [2].

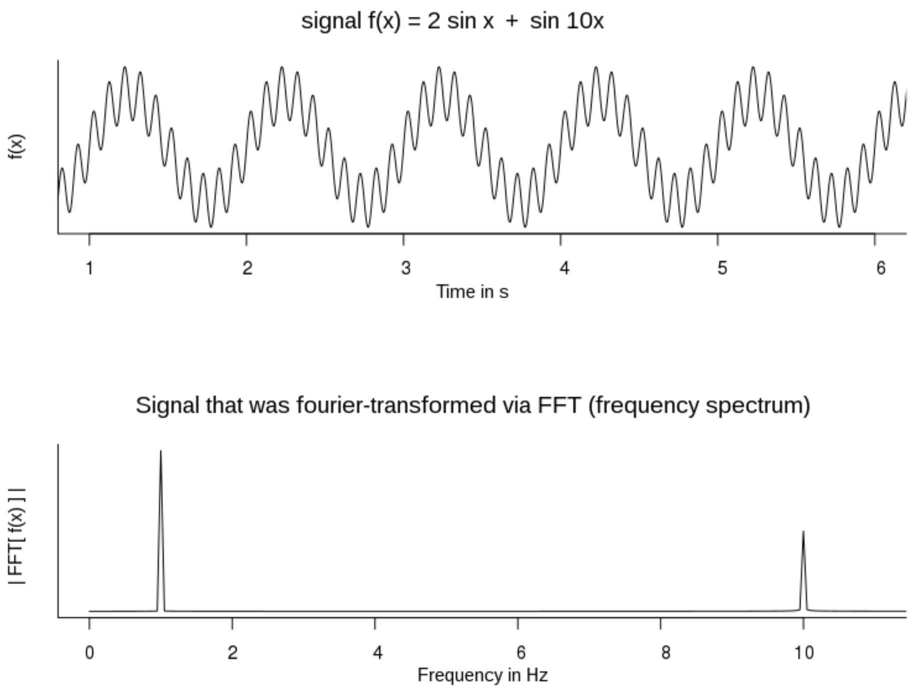

Figure 1.3: FFT of two frequency signals [2]

1.5 Motivation

The primary goal of this project is to reduce industrial motor failure on peak working days, which affects overall industry production. There were numerous articles in the newspapers about the motor's failure.

When a motor fails, the entire industry or factory comes to a halt, and the supply chain is disrupted. It costs too much to industries as production and revenue. If there are any preventive measures available to monitor the health of motors. So we know why the motors are failing and can prevent them from completely failing. The motivation behind creating a motor health monitoring system is to ensure that motors are functioning at their optimal level and to detect potential issues before they lead to significant failures. Motors are essential components in many industrial and commercial applications, and their failure can result in costly downtime, lost productivity, and even safety hazards.

Motors have different mechanical failures such as the moving parts of the motor, such as bearings, shafts, and gears. These failures can be caused by wear and tear, lack of lubrication, or misalignment of components.

By monitoring a motor's health, such as its temperature, vibration, and electrical performance, maintenance teams can detect issues early and schedule repairs or replacements as needed. This can help reduce the likelihood of unexpected downtime and extend the lifespan of the motor. Additionally, monitoring a motor's health can help identify inefficiencies or opportunities for optimization, which can lead to cost savings and improved performance.

Overall, the motivation behind creating a motor health monitoring system is to ensure the reliable and efficient operation of motors, which is crucial for many industries and applications. Additionally, a website will provide real time data of motors.

1.6 Objective

To alleviate the electrical system in today's world, rising renewable energy supplies necessitate the operation of fossil power plants in the medium and peak demand ranges. More thermo-mechanical stressing of key components like generators happens as the number of load cycles related to the medium and peak load processes increases. This is usually due to unanticipated malfunctions, machine ageing, the risk of harm, production damage, and unexpected maintenance, among other things. As a result, vibration and temperature analysis are important procedures for spinning machinery to monitor. Monitoring generator functioning and analysing detected data anomalies is an important measure to help avoid unplanned generator damage. A continuous machine status monitoring system is desired, since it can reliably and correctly communicate real-time data from the equipment to the monitoring system, allowing any anomaly in the system to be quickly identified and appropriate counteractive action performed. Successful online monitoring can aid in the prevention of unplanned trips and the improvement of outages. Different types of sensors can be employed to sense the essential parameters that must be monitored continuously. The primary goal of these initiatives is to improve overall equipment dependability and accessibility without sacrificing production, as well as to prepare for efficient maintenance. As a result, it is critical to minimise and analyse the behaviour of these vibrations in order to ensure safe and consistent operation, and numerous standards are being developed to standardise machine vibration levels.

Machines vibrate as a result of electrical and mechanical forces, vibrating at various frequencies and amplitudes throughout the system, and these vibrations may intensify over time, causing equipment mutilation. As a result, for safe and consistent operation, it is necessary to restrict and analyse the behaviour of these vibrations, and several standards for controlling vibration levels in machines have been established. Unbalance, misalignment, bearing fault, looseness, resonance, and electrical difficulties are all recognised by simulated and measured variations in the vibration frequency spectrum. The micro-controller will receive data from the accelerometer sensor in order to detect any drift in normal running frequencies.

The goal of this research is to show how vital vibration measuring is in today's mo-tors and other electrical machinery. Electric machines produce vibration in running condition. Hysteresis and mechanical movements affect the smoothness of machine. This project aims to monitor the vibration and reduce the failure rate of electrical machines. The vibration monitoring can help to prevent wear and tear in electrical machines thereby increasing its efficiency. It can also help in reducing the maintenance cost. Early fault detection is needed for protection of electrical machines. Through precise diagnostic measures, it is possible to spot the cause of vibration. The vibration pattern is altered when machines are operated in unusual conditions. In the frequency domain, the number of fluctuations and the form of irregularities can be analysed to determine the machine's malfunction. A predictive maintenance plan can be proposed based on the fault.

The most important test is vibration measurement, which will offer the user a clear picture of the machine's current state. We can anticipate the life of devices by monitoring vibrations. Machine maintenance will benefit from vibration analysis. Electrical problems can be assessed using vibration technology. Electrical glitches can be sensed from the generation of magnetic fields in machinery. These fields produce flux, which causes electromagnetic forces to be transmitted mechanically, affecting the bearings. If a machine is to perform reliably for the duration of its intended life, it must be maintained. Maintenance's job isn't to fix broken equipment, but to keep it from breaking down in the first place. The largest frequency range is seen in accelerometers that can degree frequencies from less than 1 Hz to around 30 kHz. The basic reason behind faults is stress befallen in mechanical and electrical devices. In mechanical devices faults happens due to fluctuations in loads, overloads which may cause damage to bearing of motors and breaking of rotor bar. In electrical devices faults happens due to stator winding failure i.e., short circuit. It may cause harm to complete motor.

Electric motor vibrations will produce a frequency that corresponds to that machine, which will be stored in the processor. The frequency of defects is determined by the machine's fundamental frequency. Geometrical analysis and rotational speed can be used to determine this frequency. These frequencies are usually in the low-frequency range (less than 500 Hz).

1.7 Report Organization

Chapter 1 contains the introductory topics, which details about the basic theories needed for the thorough understanding of the thesis work. Starting from the importance of maintenance, and the different maintenance types, the chapter takes us through a brief, yet interesting history of vibration analysis: both the origin and recent developments. The chapter also explains the importance of study of vibration as it can be a promising way to reduce the cost of maintenance in a simple, quick and effective way. Section 1.4 describes the objective and motivation behind the project. The next section explains the basics of vibration patterns, which contains the definition and classification of vibration patterns. The chapter is concluded by a section which describes the role of IoT in vibration monitoring. It details about the role of IoT in monitoring and how it can be used efficiently for our benefit.

Chapter 2 contains the detailed literature survey.The chapter describes various background works by proficient authors and different research papers regarding vibration,FFT, Microcontrollers and accelerometer.

Chapter 3 contains an extensive hardware research, which details about various hardware available in the market which are suitable for our project. This will contain a survey of different components like accelerometers, processors and Wi-Fi modules. To link different devices together and to transfer data within the network, various software and interface protocols must be utilised. The next section contains the software research done in order to aid the project. The third section in this chapter aims to give a summary of the different communication protocols to connect the harware to IoT.

Chapter 4 contains the prototype testing and results on different motors with different rpms.

Chapter 5 contains conclusion and future scope of the project.

Chapter 2
Literature Review

In this unit we chapter we will go through some of the literature survey and research done for Motor Health Monitoring

2.1 Background Works

The goal of the textbook indicated in [1] is to raise knowledge about vibration measurement and the tradition of vibration measurement in vibration diagnostics. While functioning, all machines vibrate to some extent, and the majority of these vibrations are unpleasant, so the challenge is to reduce them. Vibrations are only directly a working principle of particular types of devices, and they are purposely created (e.g., vibrating screeners). As a result, vibration diagnostics are uninterested in this set of equipment.

Vibration analysis using FFT is investigated experimentally in paper [9]. This is done by looking at the various working circumstances of rotating electrical devices. As a reference test, the rotating electrical machinery will be operated under normal working conditions. A series of experiments were conducted in which a variety of machine problems were purposefully introduced. In total, four tests were undertaken, with vibration data being gathered on a regular basis throughout. In the gear mesh frequency range, which is determined by the number of teeth, mechanical difficulties produce significantly lower amplitude vibration signals in a higher frequency range. Numerical data given by vibration analysis spectra offers usefulness in predictive maintenance and diagnosis of machine faults.

Paper [8] proposed a rotational machine diagnostic platform. This paper used non intrusive sensing, on-line data gaining, machine health assessment, and fault analysis in combination with diagnosis algorithms to assess the health of the machine and the severity of faults based on ISO standards, to perceive and recognise the causes of rotary machine faults with non-intrusive sensing, on-line data gaining, machine health assessment, and fault analysis. The Diagnostic Platform was created to be a valuable tool for rotary machine malfunction diagnosis.

In paper [10] a new accelerometer based on IoT was introducced. This is called MemsIO. Memsio employs a microcontroller with wireless connectivity and solid and lowpowered MEMS accelerometer. Memsio includes a huge embedded memory that saves data throughout measurements, allowing for longer observations without the risk of data loss when using wireless data transfer. It has a web browser-based user interface that allows you to apply multiple backdrops and take measurements. Two types of measurements were used to verify the sensor: lateral movement and vibration. Memsio can precisely and consistently measure in these scenarios, according to the results. Memsio takes about one work day to run, depending on the length of time it takes to perform the measurements as well as the frequency with which they are performed. Memsio's long battery life and simple installation demonstrate that it may be used for quick tests or troubleshooting. Memsio's operation period is extended by utilising cyclic measurement with deep sleep, and new possibilities with more permanent measurement needs are presented.

A more cost-effective and relaxed method of measuring vibrations from massive spinning equipment was investigated in the study [11]. Installing a connected sensor to already-running equipment with multiple spinning components is time-consuming and costly. To overcome this problem, a wireless sensor device with a MEMS accelerometer and a CPU with a WLAN module was used. The sensor unit was powered by a battery, so there was no need for wiring, and it could be used with little effort. A single sensor unit was capable of measuring acceleration in all three directions, and the data was collected from a bearing cover. As a result, only one unit was required for each bearing case. The sensor device can detect vibrations from huge rotors, according to the findings. The harmonic components and natural frequencies of the rotor might be deduced from the results. Vibrations from the bearings may also be detected. The harmonic vibration component amplitudes deviated from the true amplitudes because the measurements were taken during a continuous acceleration of a rotor and the vibration was regularly fluctuating (i.e., amplitudes measured at specified speed and load).

In publication [8] a relatively new theoretical framework for using rotor-mounted accelerometers to quantify absolute constant and transient vibratory motion of a rotor is described. By evaluating and comparing analytical and experimental results, the validity of the technique has been established. Zero-g and positional offsets require calibration prior to operating the rotor, according to an examination of error sources, and strategies to mitigate them have been presented. The innovative topology was utilised to investigate unbalanced steady state rotor data. By comparing the obtained vibration findings to eddy current displacement measurements, the results were validated. It has been discovered that without the use of timing markings, the phase and magnitude of the response can be determined directly from accelerometer data. Impact response measurements were used to learn more about accelerometer-based transient vibration sensing, with the results revealing that internal accelerometer measurements and conventional eddy current sensor data are very similar. The transformation between inertial and rotating position frames can be directly translated to variations in the spectral components of both types of sensor signals. Internal accelerometers have also been proven for rotational speed measurement, with an approach based on tracing the gravitational component in the frequency spectrum of accelerometer data.

In paper [2] baiscs of the fft is discussed.In which we learn the FFT is an algorithm used to efficiently compute the Discrete Fourier Transform (DFT) of a sequence of values. The DFT is a mathematical transformation that takes a sequence of values in the time domain and converts it into a sequence of values in the frequency domain. The possibilities of IoT for smart maintenance were demonstrated in a paper [12]. This article explained how IoT may help with rotating machinery condition monitoring and fault diagnostics using vibration analysis. Additionally, vibration detection was discussed, as well as a new generation of MEMS accelerometers. The superiority of MEMS accelerometers over historical piezoelectric accelerometers, as well as their significance in the future of smart maintenance, were highlighted in this study. This study explored making rotating equipment IoT-enabled by adding hardware to it. The hardware was conceived and constructed in accordance with the requirements. The estimation's results have been confirmed to be extremely accurate. This article validated the idea of employing IoT-enabled sensors in predictive maintenance to fill in some gaps in the commonly used techniques of rotating machinery condition monitoring. The findings can be applied to a broader range of equipment, defects, and characteristics.

Chapter 3
Hardware and Protocol Selection

In this Chapter, we will select the hardware and protocols as per our project requirements which will provide efficient results and examine the block diagram and flowchart in addition to the fundamental elements and their specifications.

3.1 Hardware Research

3.1.1 Choice of Accelerometers

The selection of accelerometers is influenced by a number of parameters, some of which are given below:

1. **Sensitivity :**The ratio of its electrical output to its mechanical input is known as sensitivity. For devices that generate their own voltage without the use of an external voltage power source, the sensitivity description is sufficient. The sensitivity of an instrument that requires an external voltage is typically expressed in terms of output voltage per unit of voltage applied per unit of displacement, velocity, or acceleration, such as millivolts per volt per g of acceleration.

2. **Amplitude Limit:** The maximum range of acceleration that the accelerometer can produce is measured by the amplitude limit.

3. **Shock Limit** is the highest amount of acceleration that the accelerometer can withstand without causing damage.

4. **Natural Frequency:** The frequency at which an undamped system with a single degree of freedom oscillates on transient displacement from its rest position is known as Natural Frequency.

5. **Resolution:** the smallest change in input that causes a change in the electrical output is called resolution. The mechanical design and the transduction element determine the resolution of an accelerometer. The resolution of the whole measurement system is frequently determined by the indicators, recording equipment, and other auxiliary equipment used with accelerometers.

6. **Amplitude Linearity** the degree of precision the accelerometr gives voltage terms when it transitions from being excited at the smallest observable acceleration levels to the greatest can be characterised as amplitude linearity. With a 1 percent variance, this accuracy is fit by its linearity.

7. **Frequency Range:** The operational frequency range during which the transducer's sensitivity does not deviate more than a certain percentage from its quoted sensitivity is known as the frequency range. The electrical or mechanical features of the transducer, as well as its associated ancillary equipment, may limit the range.

8. **Phase Shift:**The temporal delay between the mechanical input signal and the equivalent electrical output signal of the instrumentation system is known as phase shift.

More factors, such as the ones as the following, could be responsible:

- Environmental determinants (such as humidity, temperature, electromagnetic noise tolerances, etc.)
- Options for sensor installation.
- Resonant frequency that has been mounted.
- Establishing a solid foundation (isolated on non-isolated).
- Transverse sensitivity.
- Wear, dampness, and other mechanical resistance
- Dimensions .

MEMS capacitve acceleromtrs are replacing traditional piezoelectric accelerometers and other sensors in applications that previously relied on them. However, with so many accelerometers and applications available, picking the perfect one might be dif-ficult. There is no industry standard that specifies which categories an accelerometer considers to be valuable.Accelerometer grades and their applications given in ”Table 3.1” [3].

Accelerometer Grade	Main Application	Bandwidth	g-Range
Consumer	Motion, Static Acceleration	0 Hz	1g
Automative	Crash/ Stability	100 Hz	200g
Industrial	Platform Stability	5 Hz to 500 Hz	25g
Tactical	Weapons/ Craft Navigation	1 Hz	8g
Navigation	Submarine/ Craft Navigation	300 Hz	15g

Table 3.1: Accelerometer grades and their applications [3]

Given shows the categories into which accelerometers are typically classified, as well as the applications that go with them. With the rise of Industrial IoT, an emphasis

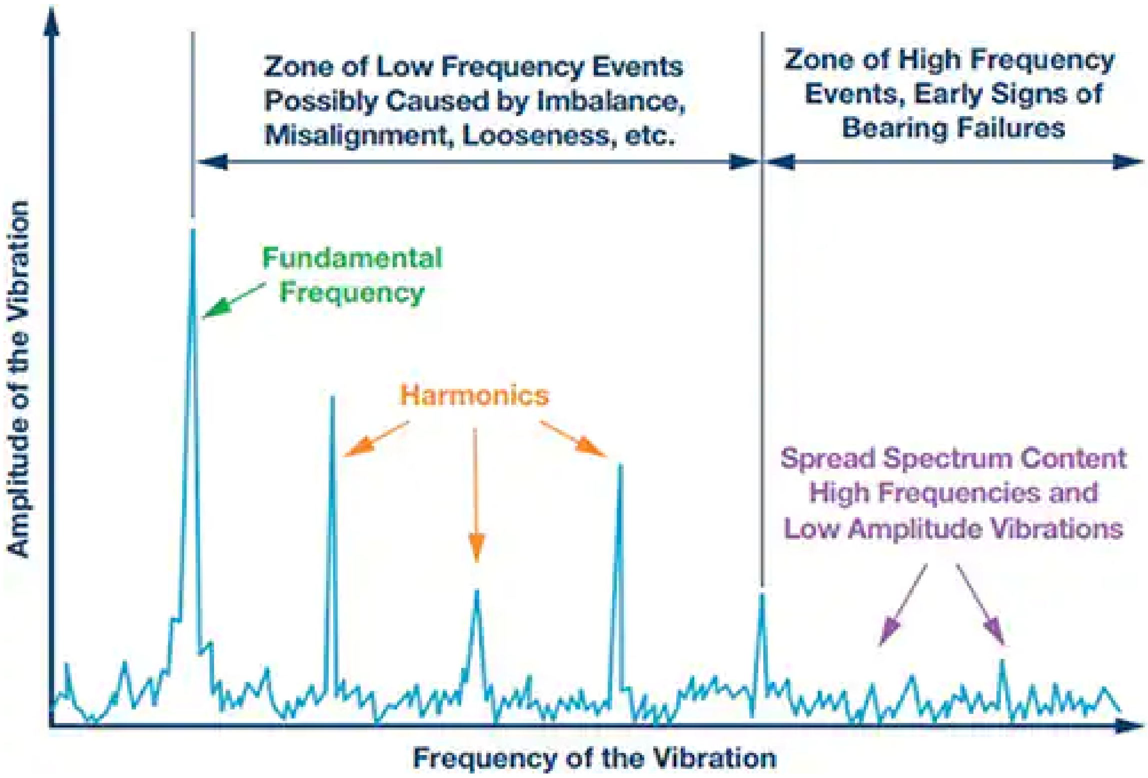

Figure 3.1: Frequency of vibration vs Amplitude plot [3]

on reducing cabling and utilising wireless, ultra-low-power solutions is becoming increasingly vital. MEMS accelerometers are preferable to piezoelectric accelerometers in terms of size, weight, power consumption, and the potential for incorporated intelligent features.Frquency of vibrations vs Amplitude plot shown in ”Figure 3.1” [3].

Collecting the vibration patterns

A sensing system to collect the vibration patterns is needed. We have various options in market now, depending on the parameter to be measured. A comparison of the three is given in the table.

As it is visible from table, accelerometers lead in performance over velocity probes and displacement transducers in terms of ease of installation, cost, frequency range and all. It is the type of sensor with least disadvantages and also is easy to use with less cost.

There are various options available in the market for an accelerometer. With the advent of technology, these accelerometers provide a lot of advantages to us. For our project we will need an accelerometer sensor. Application of different sensors is shown in ”Figure 3.2” [3].

For the project, our main aim is to reduce the overall cost. As can be seen from the ”Figure 3.2”, the options available at the lowest cost are ADXL345, ADXL343, ADXL335 and ADXL337.

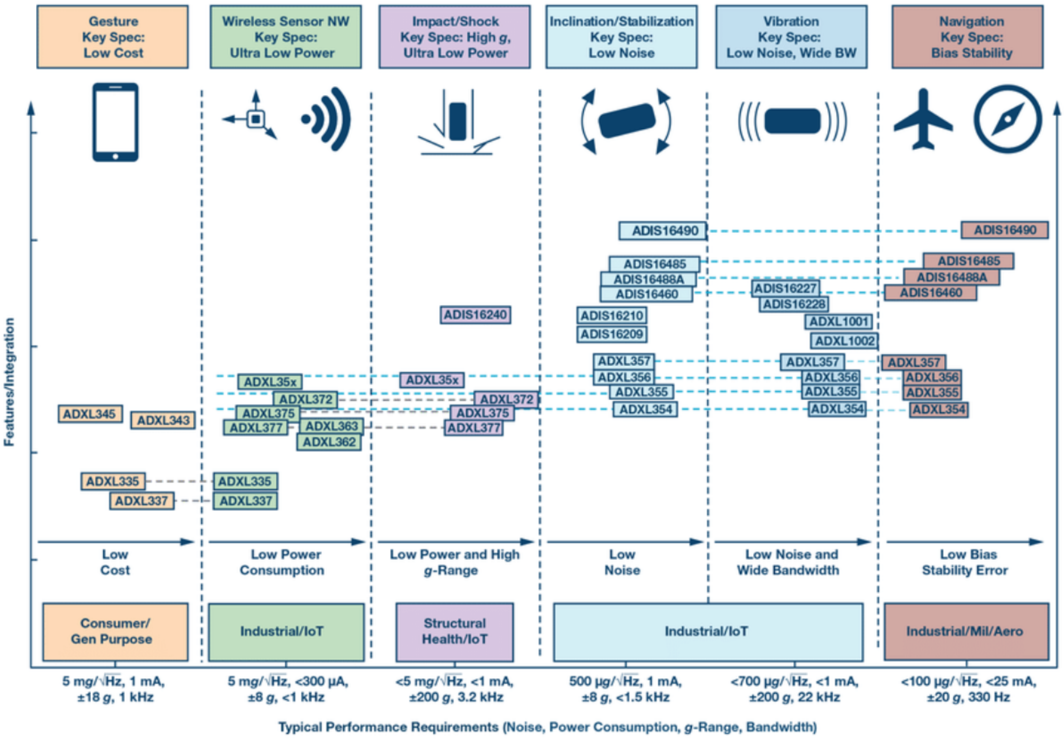

Figure 3.2: Application of different sensor [3]

3.1.2 Processing System

The two types of development kits available are those based on a microcontroller unit (MCU) and those based on a microprocessor unit (MPU) (MPU). A microprocessor and a microcontroller are two types of computer chips. A MCU is slower, more integrated, easier to programme, easier to manufacture, has no operating system, uses less power, and costs less than an MPU. An MPU, on the other hand, is more faster to programme and design, but it is also more complex. Their improved performance necessitates a large increase in power. They also employ a complete operating system, such as Linux or Android. Microcontrollers are the brains of wearable technology, remote controls, drones, and microwaves, among other things. Microprocessors, on the other hand, are the brains behind devices like computers and smart phones. In reality, by connecting a keyboard and monitor to a microprocessor development kit like the Raspberry Pi, users may use it as a stand-alone computer, allowing them to perform things like surf the web. A robot, like a person, may absorb inputs from the environment (vision, smell, touch, and so on), process the data (respond to input signals), and initiate a reaction. Similarly, an MCU/MPU may detect a wide range of analogue and digital input signals, process the data (using pre-programmed logic), and output the relevant signals. A microcontroller is a single chip with a central processing unit (CPU), memory, at least one serial port for connecting to other devices, and general-purpose input/output (GPIO) connections. Sensors, wireless radios (such as Bluetooth), actuators, motors, and electronic displays all require micro controllers to manage input/output signals from externally attached components. Micro controller and microprocessor kits, such as the Raspberry Pi, are examples of this. Previously, programming microcontrollers necessitated the use of unique serial communication interfaces, proprietary software, and specialised equipment specific to each microcontroller brand. With the arrival of open-source microcontroller development kits such as Arduino, programming microcontrollers over USB from any computer became much easier [6]. The transition from low-level hardware programming (assembly language) to high-level programming (C/C++/Python) with a common communication interface in the electronics sector was a watershed point. The cost-effectiveness of open-source development boards, as well as their ease-of-use and convenience, fueled the maker movement.Comparision between processors shown in ”figure 3.3” [4]

SPEC/BOARD	ARDUINO UNO	ESP32 8266	STM 32 f1 series (Blue -pill)	STM32 (M4) F4 series
No. of cores	1	1	1	1
Architecture	8Bit	32Bit	32Bit	32 bit(cortex M4)
CPU Frequency	16Mhz	80Mhz	72Mhz	90MHz
WiFi	NO	YES(2.4GHz)	NO	NO
Bluetooth	NO	YES	NO	NO
RAM	2KB	36KB	20KB	96KB
Fash	32KB	512Kb	64KB	512Kb
GPIO Pins	14	10	37	63
Busses	SPI,I2C,UART	SPI/I2C/UART	SPI,I2C,UART,CAN	I2C/UART/SPI/USART
ADC pins	6	1(10bit Resolution)	10(10 bit resolution)	16
DAC pins	0	0	0	0
Operating current	40mA(Average)	80mA(Average)	6mA (Average)	260uA(Average)
Price	400Rs	800Rs	500-600Rs	4000-6000Rs

Figure 3.3: Comparision between processors [4]

Important parameters about the processing system :

- **Voltage:** The voltage is the first feature to consider when selecting a development kit. The technical specification for each development will list two sorts of voltages: input voltage and operating voltage.

- **Input Voltage** The voltage supplied by an external power source to the kit (usually a wall adapter).
- **Operating Voltage:** Externally attached devices receive a regulated voltage supply from the board. While the input voltage varies from kit to kit (7-20V), the operating voltage is usually fixed at 3.3V or 5V.
- **Current:** The amount of current that can pass through each development board is controlled by an internal impedance. The restriction is usually put in place to protect the circuitry and provide adequate interface with external devices.
- **DC current per I/O pin:** If externally connected electrical components are coupled to the development board, this indicates how much current is accessible to them.
- **Clock-speed:** Clock speed is only useful as a rough indication of performance differences across similar processors. Unless the project is extremely complex or time-sensitive, clock speed is usually not a major problem. If the arrangement will be powered by batteries, however, it's best to choose microcontrollers with lower clock speeds.
- **GPIO:** General Purpose Input Output pins are included on development boards to provide additional versatility when creating electronic devices. The advantage of employing these pins for external communication is that the CPU may define the behaviour of each pin at run-time.
- **Analog or Digital pins:** GPIO pins are digital and can only be in one of two states: HIGH or LOW. They cannot be in any other state. A HIGH signal is one that is greater than 66 percent of the operational voltage, whereas a LOW signal is one that is less than that. Push buttons or relays, can be connected to digital I/O pins for instance.

3.2 Hardware Selected

As per Project requirement these compatible hardware are selected which fulfilled all the required criterion for effective and low cost vibration measurement.

3.2.1 ADXL345

The ADXL345 is a three-axis accelerometer with high resolution (13 bits), capable of measuring up to 16 g in a small, thin, low-power device. The 16-bit twos complement digital output data can be accessed via an SPI (3- or 4-wire) or I2C digital interface. The ADXL345 is an excellent choice for mobile devices. It detects both the static acceleration of gravity and the dynamic acceleration induced by motion or shock in tilt sensing applications. It can resolve inclination deviations as small as 0.25° due to its high resolution (4mg/LSB) [5]. There are a number of distinct sensor functions available. Activity and inactivity sensors detect the presence or absence of motion, as well as if the acceleration on any axis exceeds a user-defined threshold. Tap sensing is used to detect single and double taps. Free-fall sensing identifies when the gadget is falling. These routines can be connected to interrupt output pins. A 32-level FIFO can be used to store data to reduce host CPU interference. Low power modes provide sophisticated motion based power management with threshold sensing and active acceleration monitoring at extremely low power dissipation.The functional block diagram of ADXL345 shown in ”Figure 3.4” [5]. The ADXL345 is a comprehensive three-axis acceleration measurement device with a measurement range of 2 g, 4 g, 8 g, or 16 g that can be selected. It can detect both dynamic and static accelerations, such as those caused by motion or shock, allowing it to be utilised as a tilt sensor. The sensor is a micromachined polysilicon structure placed on top of a silicon wafer. Polysilicon springs hold the structure in place on the wafer’s surface and give resistance to accelerating forces. Differential capacitors, which are made up of independent fixed plates and plates attached to the moving mass, are used to measure the structure’s deflection. The differential capacitor is unbalanced when the beam deflects, resulting in a sensor output whose amplitude is proportional to acceleration. The magnitude and polarity of the acceleration are determined using phase-sensitive demodulation.Specification of ADXL345 shown in ”Figure 3.5” [5].

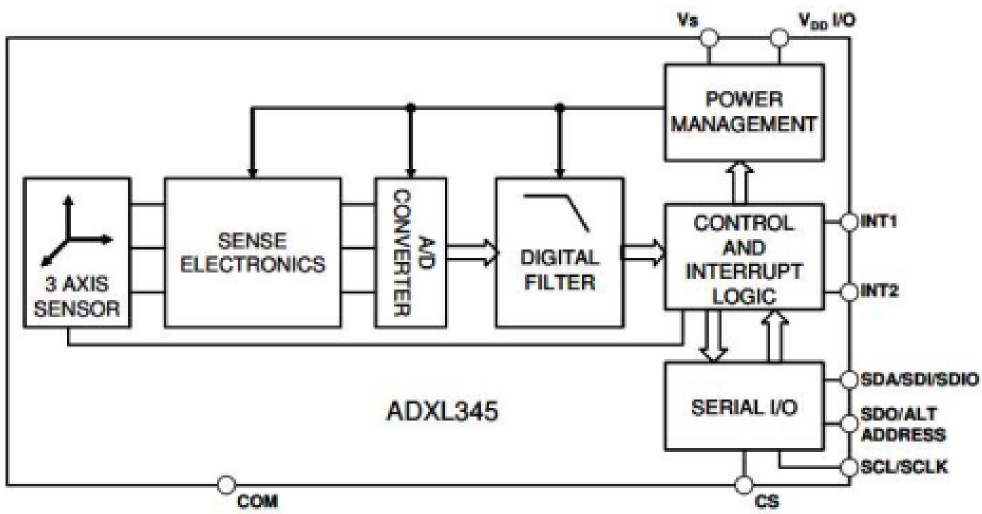

Figure 3.4: Functional block diagram of ADXL345 [5]

Parameter	**Value**
Sensor input measurement range	+/- 2,4,8,16 g
Output Resolution	10 to 13 bits
Sensitivity	232 to 286 LSB/g
Output Data Rate	0.1 to 3200 Hz
Operating Voltage Range	2.0 to 3.6 V
Supply Current	130 t0 150 uA
Operating Temperature Range	-40 to 85*C
Device Weight	20m grams

Figure 3.5: Specification of ADXL345 [5]

3.2.2 NodeMCU ESP8266

The ESP32 NodeMCU is a development board based on the ESP32 microcontroller. The ESP32 is a powerful Wi-Fi and Bluetooth-enabled microcontroller that is widely used in IoT and embedded systems. The NodeMCU is a popular development board that makes it easy to get started with the ESP32 microcontroller.The ESP32 NodeMCU shown in "Figure 3.6" [6].

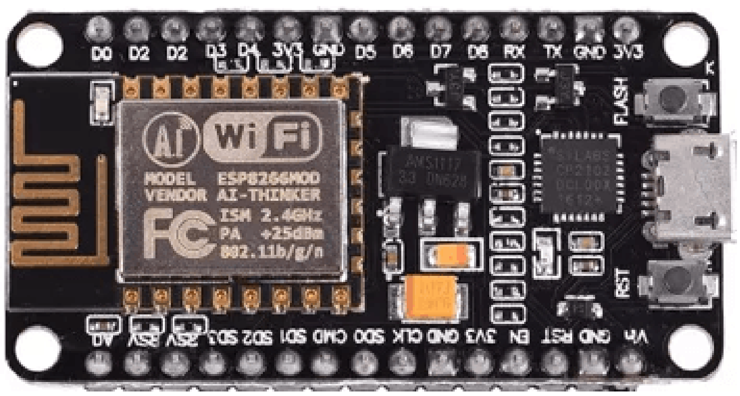

Figure 3.6: ESP32 NodeMCU [6]

Here are some key features of the ESP32 NodeMCU:

- **Processor**: The ESP32 NodeMCU features a dual-core Tensilica LX6 microprocessor with a clock speed of up to 240MHz.

- **Wi-Fi and Bluetooth:** The ESP32 NodeMCU features built-in Wi-Fi and Bluetooth connectivity, making it easy to connect to the internet and other devices.

- **GPIO Pins:** The board features 38 GPIO pins, which can be used for digital input/output, analog input, and PWM.

- **Analog-to-Digital Converter (ADC):** The ESP32 NodeMCU features a 12-bit ADC with up to 18 channels, allowing it to read analog signals from sensors and other devices.

- **Memory:** The board has 4MB of flash memory and 520KB of SRAM.

- **Power:** The ESP32 NodeMCU can be powered by a USB cable or an external power supply, and has a built-in voltage regulator to provide a stable voltage to the board.

- **Programming:** The board can be programmed using the Arduino IDE or the ESP-IDF development framework.

Overall, the ESP32 NodeMCU is a powerful and versatile development board that is suitable for a wide range of IoT and embedded systems applications.

3.2.3 ESP8266 WiFi Module

The ESP8266 is a low-cost Wi-Fi module that is popular in the hobbyist and maker communities. It was developed by the Chinese company Espressif Systems, and was released in 2014. The ESP8266 module is based on the ESP8266 System-on-Chip (SoC), which integrates a microcontroller, Wi-Fi radio, and other peripherals into a single chip [6].

The ESP8266 is widely used for Internet of Things (IoT) applications, such as home automation, smart devices, and remote sensors. It can be programmed using the Arduino IDE, as well as other programming languages and development environments. ESP8266 WiFi Module shown in ”Figure 3.7” [6].

The bandwidth of the ESP8266 Wi-Fi module depends on several factors, including the specific model of the module and the wireless standards it supports.

The most common ESP8266 module, the ESP-12, supports 802.11 b/g/n Wi-Fi standards. In theory, the maximum data rate for 802.11b is 11 Mbps, 54 Mbps for 802.11g, and up to 150 Mbps for 802.11n, depending on the number of antennas used and the channel width [6].

Figure 3.7: ESP8266-WiFiModule [6]

3.2.4 Connection between ADXL345 and ESP8266

In the "Figure 3.8"shows basic connection between the Connection between ADXL345 and NodeMCU-ESP8266 [3].

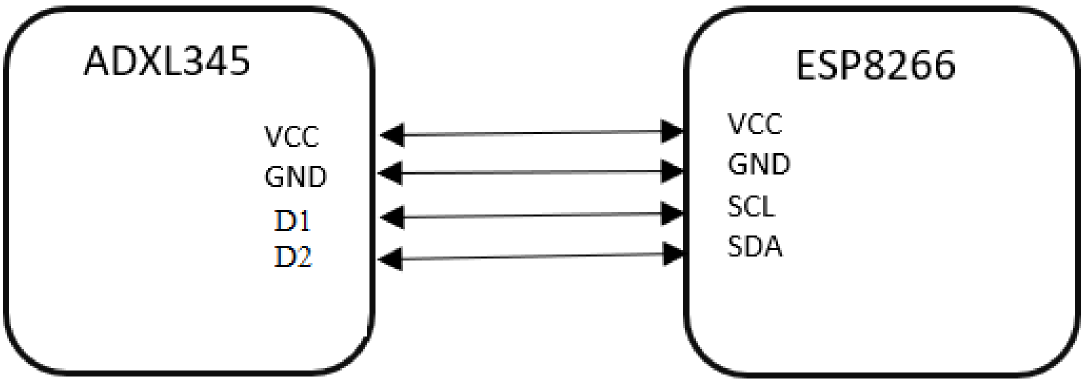

Figure 3.8: Connections between ADXL345 and NodeMCU-ESP8266 [3]

3.3 Database

A database is a collection of data that is organized and stored in a way that enables efficient retrieval, management, and manipulation of data. It can be thought of as an organized electronic filing system, where information is stored in tables or other structures that can be accessed and searched easily.

3.3.1 Firebase

Firebase is a mobile and web application development platform that was acquired by Google in 2014. It provides a variety of tools and services that enable developers to build high-quality apps quickly and easily. Some of the key features of Firebase include:

1. **Real-time Database:** Firebase provides a cloud-hosted NoSQL database that enables real-time syncing between clients and servers. This means that changes made to the database are immediately reflected on all connected devices.

2. **Authentication:** Firebase includes a robust authentication system that allows developers to easily authenticate users with email and password, phone numbers, and popular social media networks like Google, Facebook, and Twitter.

3. **Cloud Functions:** Firebase allows developers to write serverless functions that run in the cloud, without having to manage infrastructure. These functions can be triggered by events in the database or by HTTP requests [7].

4. **Cloud Storage:** Firebase provides cloud storage for user-generated content such as images, videos, and other files.

5. **Analytics:** Firebase includes a powerful analytics platform that provides insights into user behavior, app performance, and engagement.

6. **Hosting:** Firebase provides a simple and scalable hosting solution for web apps.

7. **Machine Learning:** Firebase includes machine learning tools such as ML Kit, which enables developers to integrate machine learning features into their apps with minimal code.

Overall, Firebase is a powerful and versatile platform that can help developers build high-quality apps quickly and easily. Its extensive feature set, combined with its ease of use, make it a popular choice for mobile and web app development [7].

3.4 Communication Protocol

Communication protocols are the right representations of digital message formats as well as rules. These protocols' principal purpose is to transfer data from one computer system to another. These are crucial in telecommunications networks since they deliver and receive messages on a regular basis. Error detection and repair, authentication, and signaling are all covered by these protocols. They can also explain semantics and syntax, as well as connect analog and digital communications. These protocols can be implemented in both hardware and software. So, because there are thousands of different types of communications protocols that are utilized in analog and digital communications, computer networks cannot function without them. Communication protocols allow different network devices to communicate with one another by transmitting analog signals, digital signals, various files, and data processing from one device to another. These protocols are used in telecommunications and computer networks, where proper rules are used to send data from one point to another. TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) are two of the most popular networking protocols (UDP) [12]. As shown below, there are two types of communication protocols:

- Inter System Protocol
- Intra System Protocol

3.4.1 Inter System protocol

The inter-system protocol allows two devices to communicate with each other. The communication between a computer and a microcontroller kit, for example, is accomplished using an inter bus system. The following are the most common intersystem protocol categories:

- UART Protocol
- USART Protocol
- USB Protocol

3.4.2 Intra System protocol

The Intra System Protocol is used to communicate between the two circuit board device. The following protoclos fall into various categories of intra system protocol:

- I2C PROTOCOL
- SPI PROTOCOL

I2C Protocol

I2C stands for inter-integrated circuit, and all peripherals are connected to the microcontroller using only two wires. It's a master-slave communication protocol with a unique address for each slave.The master device reads/writes the flag and instructs the address of the target slave device. Once the address is matched with the slave device, communication between the master and the slave device begins, with data being transmitted and received. When the transmitter sends 8-bit data, the slave (receiver) responds with a 1-bit acknowledgement. When the communication is complete, the stop condition is issued by the master. Philips Semiconductors was the first to develop the I2C bus. Its original purpose was to make it simple to connect the CPU to peripheral chips. Memory-mapped peripheral devices are frequently used in embedded systems to connect peripheral devices to the microcontroller. Only two wires are required to connect all peripherals to the microcontroller using I2C. These active wires, referred to as SDA and SCL, are bidirectional [10].

SPI Protocol

The serial peripheral interface is abbreviated as SPI. Motorola discovered it as one of the serial communication methods. The SPI PROTOCOL is also referred to as a 4-wire protocol. MOSI (Master Output Slave Input), MISO (Master Output Slave Input), cS (Chip Select), and SCLK are the four wires required (Serial Clock). The SPI PROTOCOL is used to send and receive data between the master and slave devices. The master uses a frequency to arrange a clock. The master then selects the slave device using the CS (Chipselect) pin. When a device's chip choose is set, that device is chosen and communication between the master and slave begins. Only one slave can be chosen at a time by the master [10].

3.5 Library Used

1. **Adafruit_ADXL345 U.h :** The Adafruit_ADXL345 is a small, low-power, 3-axis accelerometer sensor that can be used to measure acceleration in a variety of applications. It is designed to be easy to use and to integrate with microcontrollers and other electronic devices. The sensor can detect acceleration in ranges of ±2g, ±4g, ±8g, or ±16g with a high resolution of up to 13-bit. It communicates with microcontrollers over the I2C or SPI bus protocols, making it a popular choice for use in a wide range of projects [13].

   The ADXL345 is a versatile sensor that can be used in a wide range of applications, including robotics, motion sensing, gaming, and health monitoring. For example, it can be used to detect the orientation of a device, measure the impact of a collision, or track the movement of a person or object. It is also commonly used in combination with other sensors, such as gyroscopes and magnetometers, to create a more complete picture of motion and orientation.

   Adafruit, a well-known manufacturer of open-source hardware and software, has created an easy-to-use breakout board for the ADXL345 sensor, making it even more accessible to hobbyists and engineers. The board includes all of the necessary components to power the sensor and communicate with it, as well as a set of pin headers for easy connection to a microcontroller or other device. The board also includes an onboard voltage regulator, allowing it to be powered with a wide range of voltages.

Overall, the Adafruit ADXL345 is a highly versatile and easy-to-use accelerometer sensor that can be used in a wide range of applications. Its small size, low power consumption, and high accuracy make it an excellent choice for hobbyists and professionals alike.

2. **ArduinoFFT.h :** ArduinoFFT.h is a library for the Arduino platform that provides fast Fourier transform (FFT) capabilities. The FFT is an algorithm that takes a signal in the time domain and converts it into its frequency domain representation. This transformation is useful for a variety of signal processing applications, such as audio and image processing, and is widely used in fields such as engineering, physics, and mathematics.

   The ArduinoFFT.h library provides a simple and easy-to-use interface for performing FFT on signals using an Arduino board. The library supports both real and complex signals, and provides various functions for computing the FFT of different sizes. It uses the Cooley-Tukey FFT algorithm, which is a widely used FFT algorithm that has good performance for many signal sizes [13].

   One of the advantages of using ArduinoFFT.h is that it is open-source and freely available, which means that anyone can use and modify the library according to their needs. It also works well with other Arduino libraries and platforms, which makes it easy to integrate into existing projects.

   Overall, ArduinoFFT.h is a powerful and versatile library that provides fast Fourier transform capabilities for the Arduino platform. Whether you are a hobbyist or a professional, this library can help you perform signal processing tasks efficiently and effectively.

3. **ESP8266WiFi.h :**

   ESP8266WiFi.h is a library that allows you to easily connect your ESP8266 module to a Wi-Fi network. The ESP8266 module is a low-cost Wi-Fi microchip with a full TCP/IP stack and microcontroller capabilities, making it a popular choice for Internet of Things (IoT) projects. This library provides an easy-to-use interface to connect the ESP8266 to a Wi-Fi network, configure the Wi-Fi settings, and perform various network operations.

   The ESP8266WiFi.h library is included with the Arduino IDE and provides a set of functions for configuring and connecting to a Wi-Fi network. You can use the library to scan for available Wi-Fi networks, connect to a specific network, and check the connection status. The library also provides functions for configuring the module as an access point (AP) or for creating a web server.

The ESP8266WiFi.h library is designed to work with the ESP8266 module, but it can also be used with other Wi-Fi modules and boards that support the Arduino IDE. The library provides a simple and consistent interface for connecting to Wi-Fi networks, which makes it easy to use for beginners and advanced users alike. With its many features and easy-to-use interface, ESP8266WiFi.h is a powerful tool for developing Wi-Fi connected IoT projects [13].

4. **Firebase_ESP_Client.h :** Firebase_ESP_Client.h is a header file for the Arduino Firebase library that allows developers to integrate the Firebase Realtime Database and Firebase Authentication with their Arduino projects. Firebase is a backend as a service (BaaS) platform that provides various tools and services for developing mobile and web applications. It allows developers to build applications without worrying about the backend infrastructure, making the development process faster and more efficient.

   The Firebase_ESP_Client.h header file provides functions for connecting to the Firebase Realtime Database and authenticating with Firebase Authentication using an ESP8266 or ESP32-based Arduino board. It includes functions for reading and writing data to the database, listening for real-time updates, and managing user authentication. Additionally, it includes error handling functions for debugging and troubleshooting any issues that may arise during the development process [13].

   Firebase_ESP_Client.h is a crucial part of the Firebase Arduino library, and it allows developers to quickly and easily integrate Firebase into their projects. With the library, developers can leverage the power of Firebase's real-time database and authentication services to create robust and scalable applications. The library is constantly updated and maintained, ensuring that developers can access the latest Firebase features and functionality. Overall, Firebase_ESP_Client.h is a valuable tool for any developer looking to build powerful, cloud-connected Arduino projects.

5. **Addons_TokenHelper.h :** TokenHelper.h is an addon library for the programming language C++. This library provides useful tools for managing and manipulating tokens, which are a fundamental aspect of many programming languages. Tokens are the basic building blocks of a program's syntax, and they represent the different elements of the language such as keywords, identifiers, operators, and punctuation.

The TokenHelper.h library includes a set of functions and classes that can help developers with a wide range of tasks related to tokens. For example, the library includes functions for parsing a string of code and generating a sequence of tokens that represent the code's syntax. This can be particularly useful for creating programming language parsers or for performing syntax highlighting in an IDE.

Another useful feature of the TokenHelper.h library is the ability to manipulate and transform tokens. This can be particularly useful for code generation, where tokens can be modified or combined to create new code. The library includes functions for tokenizing, detokenizing, and concatenating tokens, among other operations [13].

Overall, the TokenHelper.h library is a valuable resource for any developer working with C++ or another programming language that uses tokens. Its wide range of features and functions make it a versatile tool that can help with everything from code parsing and syntax highlighting to code generation and transformation.

6. **Addons RTDBHelper.h :**

   The Addons_RTDBHelper.h is a header file that provides helper functions for working with the Real-Time Database (RTDB) in Firebase, a popular mobile and web application development platform. The RTDBHelper.h addon simplifies working with the RTDB by providing an easy-to-use interface to store, retrieve, and manage data in the database. This addon makes it easier to work with complex data structures, such as nested data or arrays, and provides a variety of functions to read and write data to the database.

   The RTDBHelper.h addon is particularly useful for developers who are working with real-time applications, such as chat applications or real-time game applications, where data is constantly changing and needs to be updated in real-time. With the RTDBHelper.h addon, developers can easily listen for changes in the database and respond to those changes in real-time, making it easier to build responsive and dynamic applications [13].

   One of the key benefits of the RTDBHelper.h addon is that it simplifies the process of working with the RTDB, which can be complex and time-consuming without the right tools. The addon provides a range of helper functions that abstract away the details of working with the RTDB, making it easier for developers to focus on building their application logic. Additionally, the RTDBHelper.h addon is open-source and community-driven, meaning that developers can contribute to its development and improve its functionality over time.

Chapter 4
Testing and Results

After every hardware selection and software developement we have to do connections between the ADXL345 and ESP32 NodeMCU and calculate FFT in ESP32 NodeMCU and send data to the Firebase server using ESP8266 WiFiModule then we take data form firebase server and plot the FFt at website.

4.1 Hardware connection

Symbolization of hardware and software connection shown in given ”Figure 4.1”.

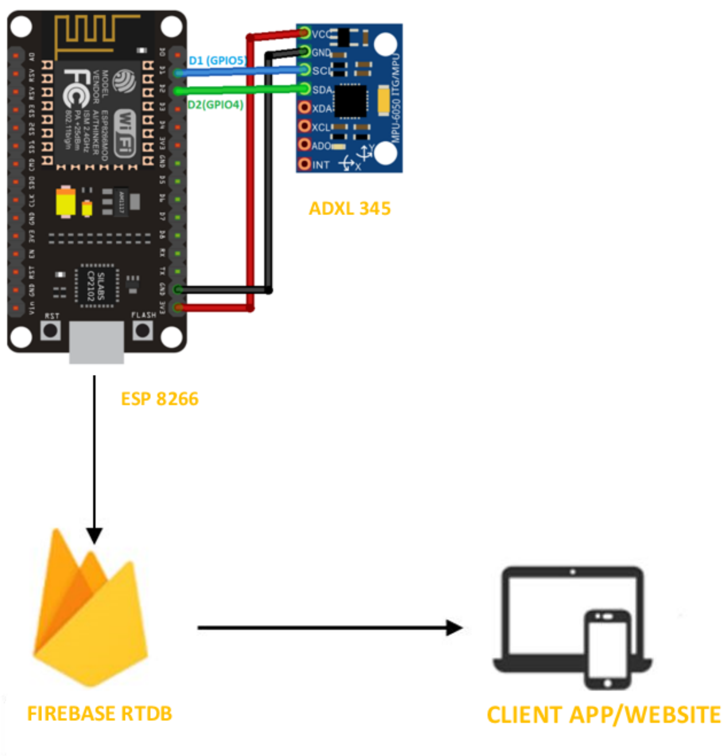

Figure 4.1: Hardware and Software connections

Hardware connection refers to the process of physically linking various hardware components of a computer system together in order to allow data and information to flow between them. This can include connecting devices such as monitors, keyboards, printers, and external hard drives to a computer, as well as connecting internal components such as the motherboard, processor, RAM, and hard drive.

Making sure that hardware components are properly connected is essential to ensuring that a computer system functions correctly. Poor or faulty connections can lead to data loss, system errors, and other problems. It's also important to make sure that connections are made securely and that cables are not damaged or frayed, as this can cause issues with data transmission and potentially damage the hardware.

4.1.1 ADXL345 to NodeMCU ESP8266

The ADXL345 is a digital accelerometer that can communicate using the I2C or SPI communication protocols. The ESP8266 is a Wi-Fi enabled microcontroller that can also communicate using these protocols.Connection between ADXL345 and NodeMCU ESP8266 is shown in "Figure 4.2" [6].

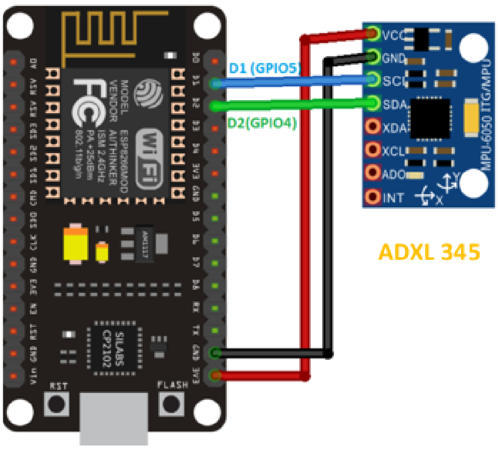

Figure 4.2: Connection between ADXL345 and NodeMCU ESP8266 [6]

Here are the steps to connect the ADXL345 to the ESP8266:

1. Connect the ADXL345 to the ESP8266 using either I2C or SPI interface. For I2C, connect the ADXL345 SDA pin to the ESP8266 GPIO2 pin and SCL pin to GPIO14 pin. Connect the VDD and GND pins to the ESP8266's 3.3V and GND pins, respectively.

2. For SPI, connect the ADXL345 SCK pin to the ESP8266 GPIO14 pin, MOSI pin to GPIO13, MISO pin to GPIO12, and CS pin to GPIO15. Connect the VDD and GND pins to the ESP8266's 3.3V and GND pins, respectively.

3. Connect the ESP8266 to your computer using a USB cable and open the Arduino IDE.

4. Install the necessary libraries for the ADXL345 and ESP8266. For the ADXL345, you can use the Adafruit_ADXL345 library and for the ESP8266, you can use the

ESP8266WiFi and either the Wire or SPI library depending on which interface you choose to use.

4.1.2 NodeMCU ESP8266 to Firebase

To connect a NodeMCU ESP8266 to Firebase, you can use the Firebase Realtime Database API to send and receive data. Connection between NodeMCU ESP8266 and Firebase is shown in ”Figure 4.3” [6].

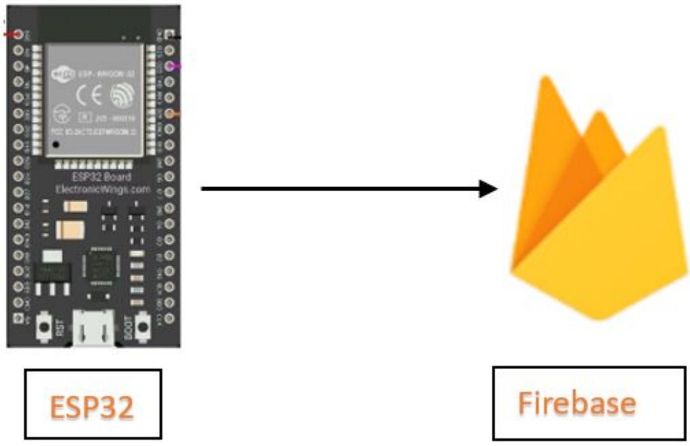

Figure 4.3: Connection between NodeMCU ESP8266 and Firebase [6]

Here are the steps to get started:

1. Create a Firebase project and enable the Real time Database. You can do this by going to the Firebase Console, creating a new project, and then navigating to the Real time Database tab.

2. Obtain the Firebase Real time Database API key, which you will use to authenticate your Node MCU ESP8266 when communicating with Firebase. To do this, go to the Project Settings page in the Firebase Console, click on the Service Accounts tab, and then click on the Generate New Private Key button to download a JSON file containing your API key.

3. Install the Firebase Arduino library. This library provides an interface for interacting with Firebase from an Arduino-compatible device, such as the Node MCU ESP8266. You can download the library from the Arduino Library Manager or from the Firebase GitHub repository.

4. Configure your Node MCU ESP8266 to connect to your Wi-Fi network. You will need to provide your Wi-Fi SSID and password in your code.

5. Initialize the Firebase library in your code and authenticate with your API key. You will need to provide the name of your Firebase project, your API key, and your Wi-Fi credentials in your code.

6. Use the Firebase API to send and receive data from the Realtime Database. You can use functions like Firebase.RTDB.set() to set data.

4.1.3 Firebase To Website

Firebase is a cloud-based platform that provides various services for building web and mobile applications. Connection between Firebase and Website is shown in ”Figure 4.4” [7].

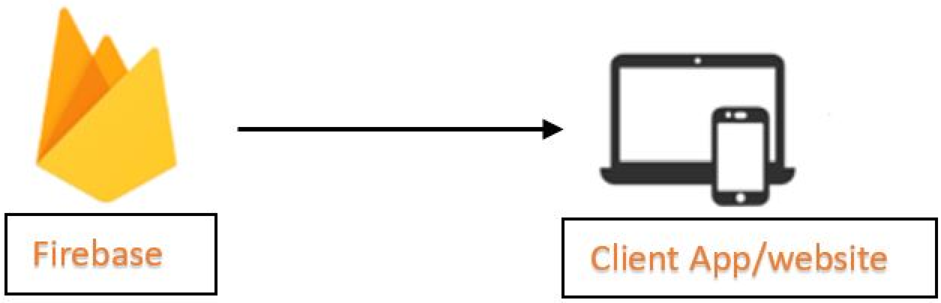

Figure 4.4: Connection between Firebase and Website [7]

If you want to use Firebase with your website, you can follow these steps:

1. Create a new Firebase project: In the Firebase console, click on ”Add Project” and follow the prompts to create a new project. Give your project a name and select your preferred region.

2. Set up Firebase hosting: Firebase hosting allows you to host your website files on Firebase servers. To set up hosting, go to the Hosting section of the Firebase console and follow the prompts to configure your hosting settings.

3. Add Firebase to your website: To use Firebase services in your website, you will need to add the Firebase JavaScript SDK to your website’s code. You can do this by including the Firebase SDK script in your HTML file or by using a package manager like NPM.

4. Use Firebase services: Once you have added the Firebase SDK to your website, you can start using Firebase services like Firebase Authentication, Cloud Firestore, Cloud Storage, and more. You can use these services to add user authentication, store and retrieve data, and more.

4.2 Steps of Testing process

1. First we mounted ADXL345 on selected motor.
2. Receive data from ADXL35 and store in memory of ESP8266 NodeMCU.
3. Calculate FFT in ESP8266 NodeMCU of stored ADXL345 data.
4. Using ESP8266 WiFi-Module share the FFT data to the Firebase
5. Take data from the Firebase and plot the FFT graph at Website.

4.3 Testing for Grinding Motor

Specification of motor and hardwares related to testing#1 are given in ”Table 4.1”.

1.	Type of Motor	Grinding Motor
2.	Condition of Motor	Good
3.	Sampling rate of ADXL345	500 Hz
4.	Total Samples	2048 Samples

Table 4.1: Specification of motor and hardwares related to testing for grinding motors

Placement of ADXL345 on grinding Motor shown in ”Figure 4.5”.

Figure 4.5: ADXL345 mounted on motor

4.3.1 Testing at 640 RPM of Grinding Motor

After Plotting FFT of the ADXL345 data on website we get following results are shown in "Figure 4.6".

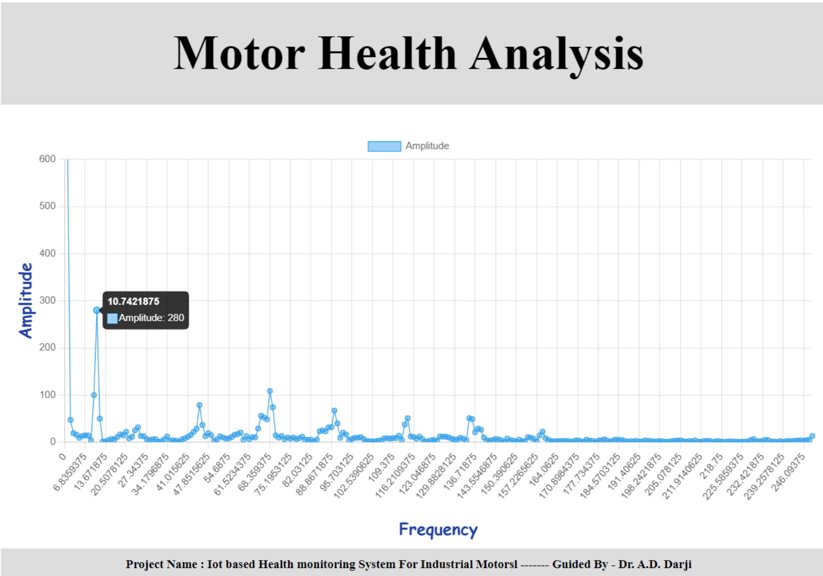

Figure 4.6: FFT of the ADXL345 data on Website for 640 RPM

Sampling rate = 500 Hz
Total No. of Samples = 2048
Motor RPM = 640 RPM
Natural Frequency from FFT = 10.74Hz
Predicted rpm = 10.74*60 = 644.4 RPM

4.3.2 Testing at 1100 RPM of Grinding Motor

After Plotting FFT of the ADXL345 data on website we get following results are shown in "Figure 4.7".

Sampling rate = 500 Hz
Total No. of Samples = 2048
Motor RPM = 1100 RPM
Natural Frequency from FFT = 18.55 Hz
Predicted rpm = 18.55*60 = 1113 RPM

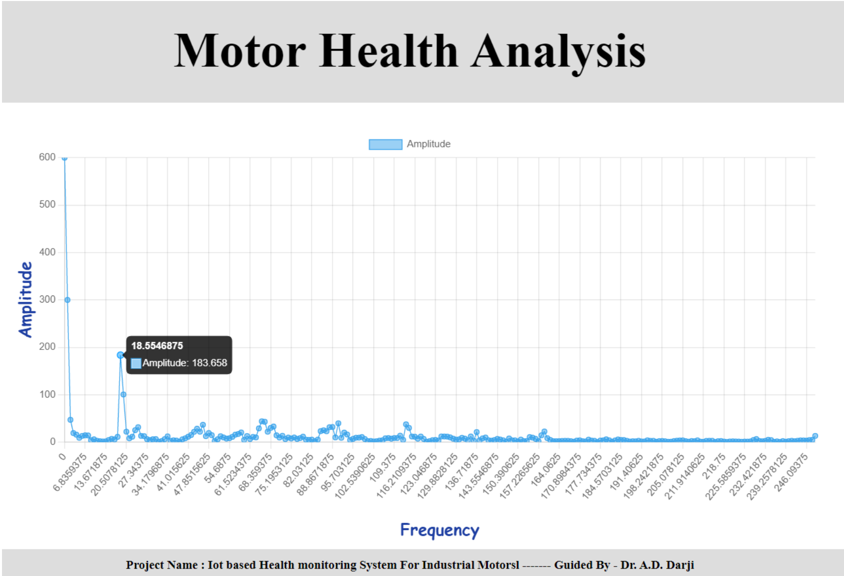

Figure 4.7: FFT of the ADXL345 data on Website for 1100 RPM

4.3.3 Testing at 1400 RPM of Grinding Motor

After Plotting FFT of the ADXL345 data on website we get following results are shown in "Figure 4.8".

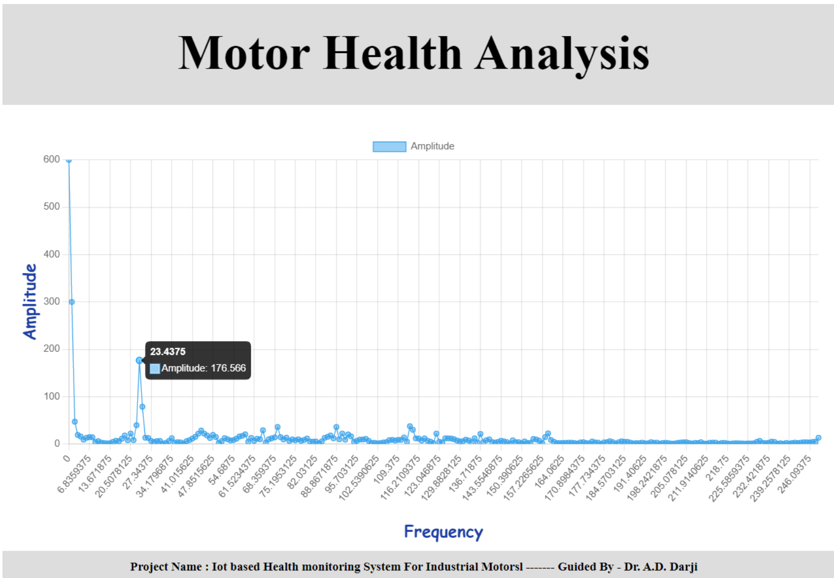

Figure 4.8: FFT of the ADXL345 data on Website for 1400 RPM

Sampling rate = 500 Hz
Total No. of Samples = 2048
Motor RPM = 1350 RPM;
Natural Frequency from FFT = 22Hz

Predicted rpm = 23.44*60=1406.4 RPM

4.3.4 Analysis

Analysis of Outputs frequencies of Grinding Motor shown in ”Table 4.2”.

S.R No.	RPM of Motor	Observed Frequency	Fundamental Fequency
1.	640	10.74 Hz	10.67 Hz
2.	1100	18.55 Hz	18.5Hz
3.	1400	23.44 Hz	23.33

Table 4.2: Analysis of Outputs frequencies of Grinding Motor

4.4 Testing for Motor with Wobbly Shaft

Specification of motor and hardwares related to testing#1 are given in ”Table 4.3”.

1.	Type of Motor	Motor with Wobbly Shaft
2.	Condition of Motor	Fine
3.	Sampling rate of ADXL345	500 Hz
4.	Total Samples	2048 Samples

Table 4.3: Specification of motor and hardwares related to testing for Motor with Wobbly Shaft

Placement of ADXL345 on Motor with wobbly shaft shown in ”Figure 4.9”.

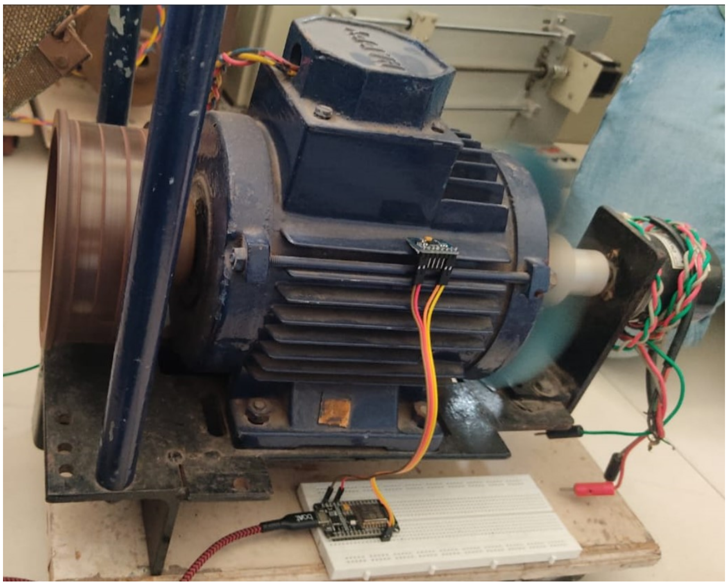

Figure 4.9: ADXL345 mounted on motor with wobbly shaft

4.4.1 Testing at 690 RPM of Motor with Wobbly Shaft

After Plotting FFT of the ADXL345 data on website we get following results are shown in ”Figure 4.10”.

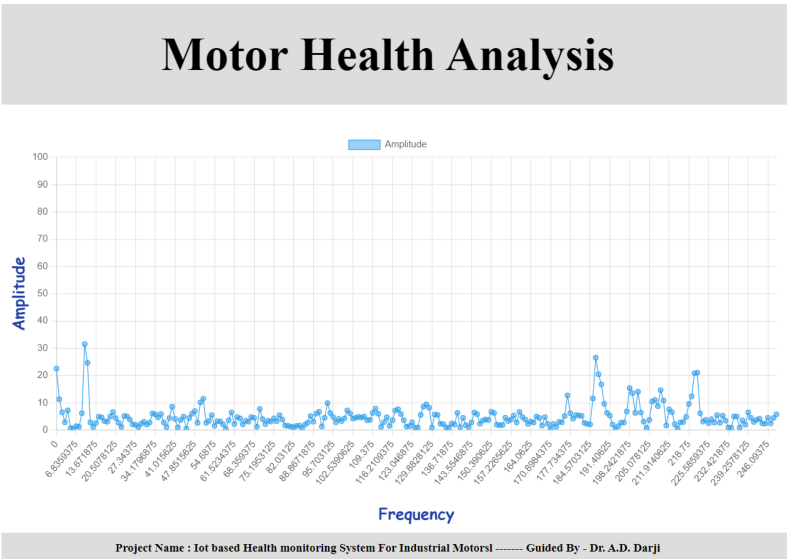

Figure 4.10: FFT of the ADXL345 data on Website for 690 RPM

Sampling rate = 500 Hz;
Total No. of Samples = 2048
Motor RPM = 690 RPM;

Natural Frequency from FFT = 11.71 Hz
Predicted rpm = 11.71*60 = 702.6 RPM

4.4.2 Testing at 1680 RPM of Motor with Wobbly Shaft

After Plotting FFT of the ADXL345 data on website we get following results are shown in "Figure 4.11".

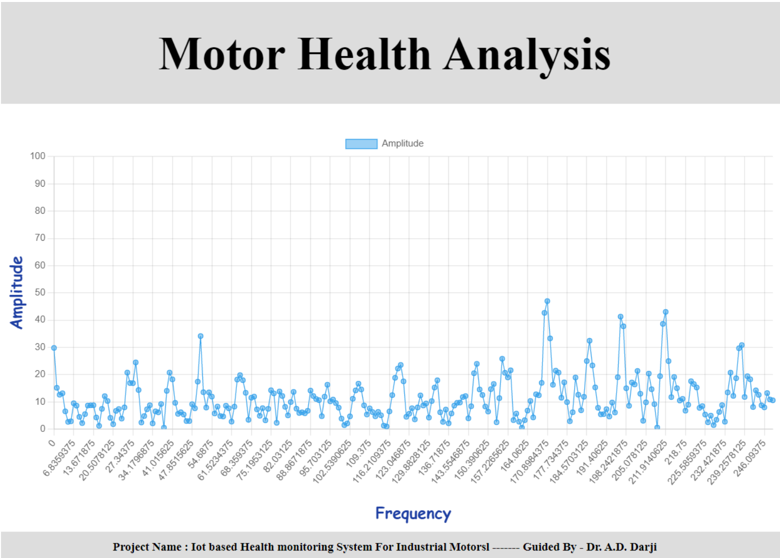

Figure 4.11: FFT of the ADXL345 data on Website for 1680 RPM

Sampling rate = 500 Hz;
Total No. of Samples = 2048
Motor RPM = 1680 RPM;
Natural Frequency from FFT = Not identifiable
Predicted rpm = Not identifiable

4.4.3 Analysis

Analysis of Outputs frequencies of Grinding Motor shown in "Table 4.4".

S.R No.	RPM of Motor	Observed Frequency	Fundamental Fequency
1.	690	11.71 Hz	11.5 Hz
2.	1680	-	28 Hz

Table 4.4: Analysis of Outputs frequencies of Grinding Motor

4.5 Testing for Table Fan Motor

Specification of motor and hardwares related to testing#1 are given in ”Table 4.5”.

1.	Type of Motor	Table Fan Motor
2.	Condition of Motor	Good
3.	Sampling rate of ADXL345	500 Hz
4.	Total Samples	2048 Samples

Table 4.5: Specification of motor and hardwares related to testing for Table fan motor

Placement of ADXL345 on Table fan Motor shown in ”Figure 4.12”.

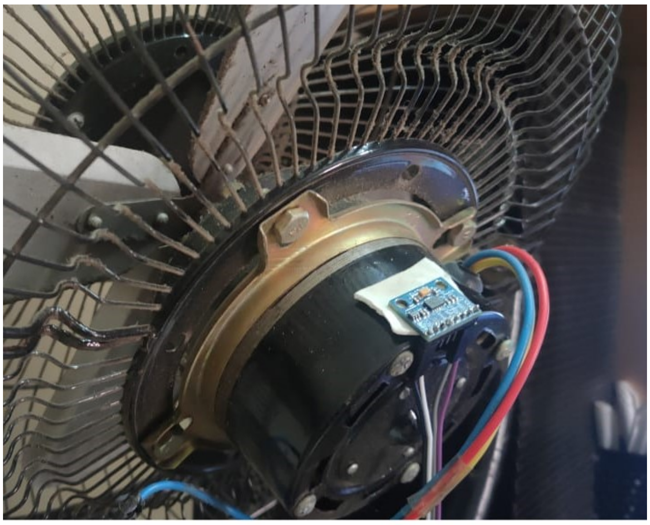

Figure 4.12: ADXL345 mounted on Table fan motor

4.5.1 Testing at 1210 RPM of Table Fan Motor

After Plotting FFT of the ADXL345 data on website we get following results are shown in ”Figure 4.13”.

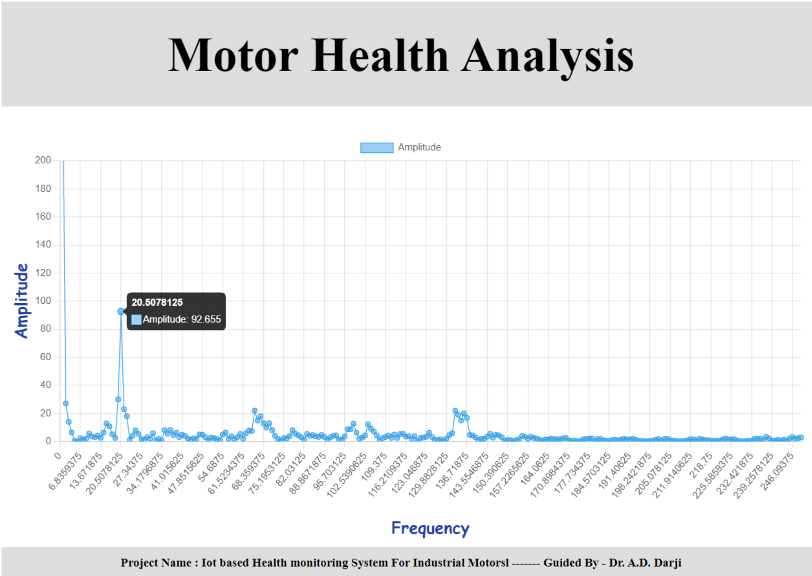

Figure 4.13: FFT of the ADXL345 data on Website for 1210 RPM

Sampling rate = 500 Hz
Total No. of Samples = 2048
Motor RPM = 1210 RPM
Natural Frequency from FFT = 20.5 Hz
Predicted rpm = 20.5*60 = 1230 RPM

4.5.2 Testing at 1500 RPM of Table Fan Motor

After Plotting FFT of the ADXL345 data on website we get following results are shown in "Figure 4.14". **Sampling rate** = 500 Hz
Total No. of Samples = 2048
Motor RPM = 1500 RPM
Natural Frequency from FFT = 25.39Hz
Predicted rpm = 25.39*60 = 1523.4 RPM

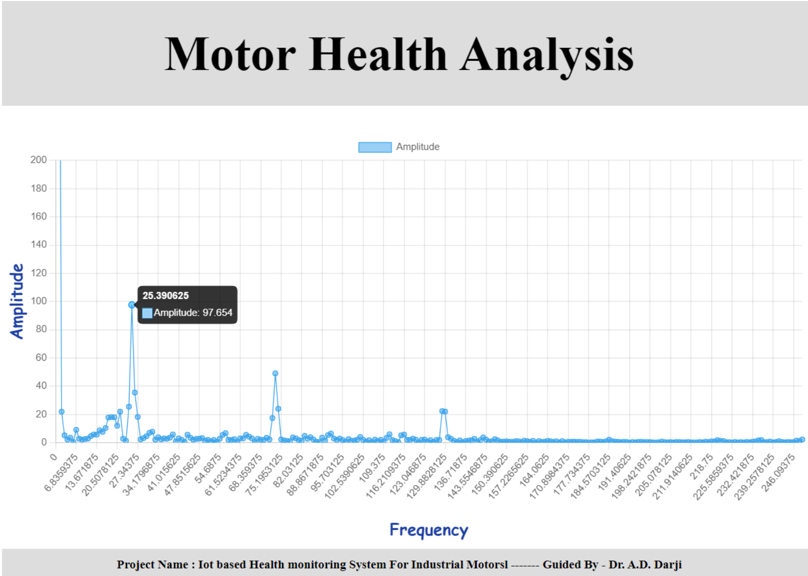

Figure 4.14: FFT of the ADXL345 data on Website for 1500 RPM

4.5.3 Testing at 1800 RPM of Table Fan Motor

After Plotting FFT of the ADXL345 data on website we get following results are shown in "Figure 4.15".

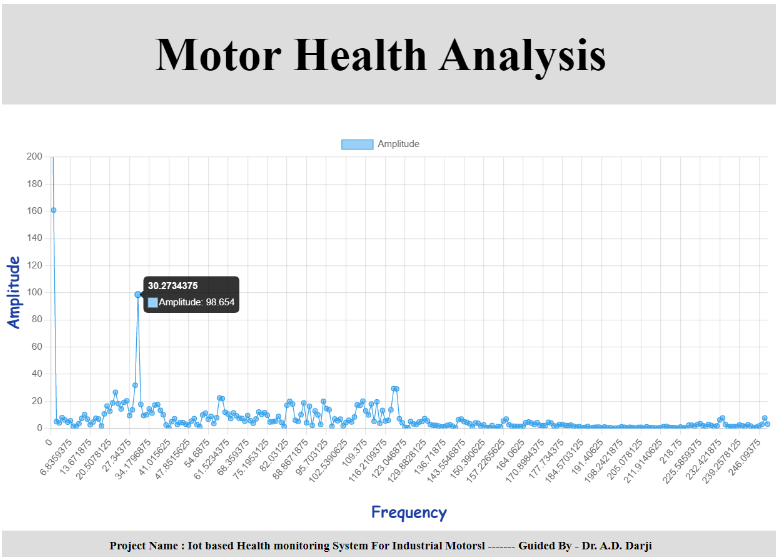

Figure 4.15: FFT of the ADXL345 data on Website for 1800 RPM

Sampling rate = 500 Hz
Total No. of Samples = 2048
Motor RPM = 1800 RPM

Natural Frequency from FFT = 30.27 Hz
Predicted rpm = 30.27*60 =1816.4 RPM

4.5.4 Analysis

Analysis of Outputs frequencies of Table Fan Motor shown in ”Table 4.6”.

S.R No.	RPM of Motor	Observed Frequency	Fundamental Fequency
1.	1210	20.5 Hz	20.16 Hz
2.	1500	25.39 Hz	25 Hz
3.	1800	30.27 Hz	30 Hz

Table 4.6: Analysis of Outputs frequencies of Table Fan Motor

4.6 Comparison between Available ideas and Proposed idea

Comparison between two available ideas in the market and idea proposed by us shown in ”Table 4.7”.

	Fluke 810 Vibration Tester	Fluke 805 Vibration Meter	Proposed idea
Acceleration Limit	+500 g to -500g	+50 g to -50g	+16g to -16g
Sampling rate	2.5 kHz to 50 kHz	2 kHz to 10 kHz	0.10 Hz to 3.2 kHz
Motor Rotational Speed Range	200 rpm to 12000 rpm	200 rpm to 10000 rpm	30 rpm to 10000 rpm
Market Price	1,71,250 Rs.	1,57,800 Rs.	Approx. 2000 Rs.

Table 4.7: Comparison between available ideas and proposed idea

*For this proposed model we have set sampling rate at 500 Hz which allows us to measure maximum rpm of 7500.Which enough for motors we dealing with.

Chapter 5
Conclusion & Future Scope

5.1 Conclusion

In conclusion, the motor health monitoring system integrated with IoT technology and a user interface that includes on-demand requests for maintenance and continuous updates is a significant advancement in the industrial sector. By continuously monitoring the motor's health, predicting potential issues, and enabling on-demand maintenance requests, this system reduces downtime and maintenance costs while improving productivity and worker safety. The integration of IoT technology allows for real-time data monitoring and analysis, providing timely insights for better decision-making. The system's user interface allows for seamless communication between operators and maintenance personnel, enabling prompt maintenance requests and quick issue resolution.vibration monitoring is a crucial process in predicting and avoiding machine failure, as it can provide early detection of faults and save on maintenance costs. This project focuses on utilizing IoT integration and hardware, such as accelerometers and microcontrollers, to continuously monitor machine status and analyze vibration patterns. By implementing a cost-effective approach, this project aims to provide a practical solution for industries to improve machine health and safety. The Internet of Things offers vast opportunities and benefits for the manufacturing industry, and with the integration of vibration monitoring, businesses can take proactive measures to prevent downtime and reduce costs.
Additionally, the integration of vibration monitoring with IoT technology enables industries to transition towards a more predictive maintenance approach. By continuously monitoring machine health and analyzing vibration patterns, businesses can identify potential issues before they result in catastrophic failures, allowing for scheduled maintenance and repairs, minimizing downtime, and extending the machine's lifespan. This can ultimately lead to increased productivity and profitability for the industry.

Furthermore, vibration monitoring can also contribute to improving worker safety. Machines that operate with excessive vibration can produce hazardous working conditions, increasing the risk of accidents and injuries. By detecting and addressing vibration issues, industries can maintain a safe working environment for their employees, reducing the likelihood of workplace accidents and injuries.

Vibration monitoring integrated with IoT technology is a valuable tool for industries to improve machine health, reduce maintenance costs, and increase productivity and profitability. By utilizing cost-effective hardware, businesses can continuously monitor machine health and predict potential failures, enabling them to transition towards a more proactive and predictive maintenance approach. This not only benefits the industry but also contributes to a safer working environment for employees.

5.2 Future Scope

The future scope of this system involves improving the predictive maintenance capabilities by incorporating machine learning algorithms and predictive analytics, which can provide more accurate predictions of motor failures. Furthermore, the system's capabilities can be expanded to include remote control and diagnostics, allowing for remote troubleshooting and issue resolution, further improving the motor's reliability and reducing downtime. Overall, the motor health monitoring system integrated with IoT technology and a user interface has enormous potential for the industrial sector, paving the way for a safer, more efficient, and cost-effective future. Machine learning (ML) and artificial intelligence (AI) are being used in the health monitoring system for motors in a variety of ways to improve efficiency, prevent failures, and optimize performance.

The field of motor health monitoring systems using IoT is rapidly evolving and there are numerous potential future developments. Here are some possibilities:

1. Enhanced real-time monitoring: Motor health monitoring systems can benefit from enhanced real-time monitoring capabilities, such as the ability to detect and predict potential issues before they occur. This can be achieved through the use of advanced analytics and machine learning algorithms.

2. Integration with other systems: Motor health monitoring systems can be integrated with other IoT systems, such as building automation and energy management systems, to optimize energy consumption and reduce maintenance costs.

3. Predictive maintenance: Predictive maintenance is an emerging field that uses data analysis to predict when maintenance is required. Motor health monitoring systems can use machine learning algorithms to predict motor failures and schedule maintenance before the motor fails.

4. Remote monitoring: Motor health monitoring systems can enable remote monitoring, allowing technicians to diagnose and troubleshoot issues from a central location. This can reduce the need for onsite visits and improve response times.

5. Condition-based maintenance: Condition-based maintenance (CBM) is a maintenance strategy that uses real-time data to determine the condition of equipment and schedule maintenance accordingly. Motor health monitoring systems can enable CBM, allowing maintenance to be performed only when necessary, reducing downtime and

maintenance costs.

Overall, the future of motor health monitoring systems using IoT is bright, with the potential to improve motor performance, reduce maintenance costs, and increase equipment reliability.

References

[1] A. Bilošová and J. Biloš, “Vibration diagnostics,” *VŠB-TU, Ostrava*, 2012.

[2] D. Kurnia, M. N. Hakim, I. S. Nurmala *et al.*, “Implementation of fft in industrial induction motor monitoring via mobile phone,” *Journal of Applied Science and Advanced Engineering*, vol. 1, no. 1, pp. 5–10, 2023.

[3] A. Datasheet and A. Devices, “Url: http://www. analog. com/media/en/technical-documentation/data-sheets,” *ADXL345. pdf (time of request: 27.08. 17)*.

[4] V. Gaikwad, M. Doiphode, M. Gatke, G. Moona, C. Gavit, and P. Ghuge, “Self-regulated robotic illuminant using nodemcu (esp8266),” in *2023 International Conference on Emerging Smart Computing and Informatics (ESCI)*. IEEE, 2023, pp. 1–4.

[5] A. POWER, “Data sheet adxl345.”

[6] E. Datasheet, “Esp8266 serial esp-01 wifi wireless,” *ESP8266 Serial Esp-01 WIFI Wireless*, p. 12, 2004.

[7] L. Moroney and L. Moroney, “The firebase realtime database,” *The Definitive Guide to Firebase: Build Android Apps on Google’s Mobile Platform*, pp. 51–71, 2017.

[8] S. Jiménez, M. O. Cole, and P. S. Keogh, “Vibration sensing in smart machine rotors using internal mems accelerometers,” *Journal of Sound and Vibration*, vol. 377, pp. 58–75, 2016.

[9] S. Patil and J. Gaikwad, “Vibration analysis of electrical rotating machines using fft: A method of predictive maintenance,” in *2013 fourth international conference on computing, communications and networking technologies (ICCCNT)*. IEEE, 2013, pp. 1–6.

[10] I. Koene, V. Klar, and R. Viitala, “Iot connected device for vibration analysis and measurement,” *HardwareX*, vol. 7, p. e00109, 2020.

[11] I. Koene, R. Viitala, and P. Kuosmanen, “Internet of things based monitoring of large rotor vibration with a microelectromechanical systems accelerometer,” *IEEE Access*, vol. 7, pp. 92 210–92 219, 2019.

[12] A. Khademi, F. Raji, and M. Sadeghi, “Iot enabled vibration monitoring toward smart maintenance,” in *2019 3rd International Conference on Internet of Things and Applications (IoT)*. IEEE, 2019, pp. 1–6.

[13] J. Purdum and J. Purdum, “Arduino libraries,” *Beginning C for Arduino: Learn C Programming for the Arduino*, pp. 277–298, 2015.

Appendix A
Adxl345 through esp32 8266

A.1 FFT simulation on website

```
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_ADXL345_U.h>
#include "arduinoFFT.h"
#include <ESP8266WiFi.h>
#include <Firebase_ESP_Client.h>
#include "addons/TokenHelper.h"
#include "addons/RTDBHelper.h"

#define API_KEY "AIzaSyByTWjYsnZQ4zSUH7ZEYJabD5XQxSppE3w"
#define DATABSE_URL "https://motor-health-analysis-default-rtdb.fire
char ssid[] = "Dinesh";
char password[] = "18032001";

Adafruit_ADXL345_Unified accel;
arduinoFFT FFT = arduinoFFT();

FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;
FirebaseJson json;
FirebaseJsonArray arr;

#define SCL_INDEX 0x00
#define SCL_TIME 0x01
#define SCL_FREQUENCY 0x02
#define SCL_PLOT 0x03

unsigned long sendDataPrevMillis = 0;
bool signupOK = false;

const int samples = 512;
int samplingFrequency = 493;
```

```
unsigned long StartTime;

double xReal[samples];
double xImag[samples];
//float yss[512];
//float zss[512];




void setup() {
  // put your setup code here, to run once:
  pinMode(LED_BUILTIN, OUTPUT);

  Serial.begin(19200);
  ///////////////////////////////////////////////////
  if(!accel.begin())
  {
    /* There was a problem detecting the ADXL345 ... check your connectio
    Serial.println("Ooops, no ADXL345 detected ... Check your wiring!");
    while(1);
  }

  /* Set the range to whatever is appropriate for your project */
  accel.setRange(ADXL345_RANGE_4_G);
  accel.setDataRate(ADXL345_DATARATE_400_HZ);
  /* Display some basic information on this sensor */


  //////////////////////////////////////////////////////

  // Connect to Wi-Fi/////////////////////////////////
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi..");
  }
```

```
  // Print ESP32 Local IP Address
  Serial.println(WiFi.localIP());

  /////////////////////////////////////////////////////

  config.api_key = API_KEY;
  config.database_url = DATABSE_URL;
  if(Firebase.signUp(&config, &auth, "", "")){
    Serial.println("signUp OK");
    signupOK = true;
  }else{
    Serial.printf("%s\n", config.signer.signupError.message.c_str())
  }

  config.token_status_callback = tokenStatusCallback;
  Firebase.begin(&config, &auth);
  Firebase.reconnectWiFi(true);


}

void loop() {
  // put your main code here, to run repeatedly:

  //taking data from the adxl////////////////////////////////////
  int i =0;
    digitalWrite(LED_BUILTIN, LOW);
  StartTime=millis();
  while(i<samples){
    sensors_event_t event;
    accel.getEvent(&event);

      xReal[i]=event.acceleration.x;
      xImag[i] = 0.0;
      //yss[i]=event.acceleration.y;
      //zss[i]=event.acceleration.z;
      i=i+1;
      if(millis()-StartTime==1000){
```

```
        samplingFrequency=i;

      }

  }
  digitalWrite(LED_BUILTIN, HIGH);
  Serial.print("Sampling Frequency :-> ");Serial.println(samplingFrequenc
  delay(1000);




///////////////////////////////////////////////////////////////////////////////
  if(Firebase.ready() && signupOK && (millis()-sendDataPrevMillis>5000||
    sendDataPrevMillis = millis();



      json.add("Sampling Frequency", samplingFrequency);
      json.add("Samples", samples);
      Serial.printf("Sending Sampling Frequency %s\n", Firebase.RTDB.set(
      delay(2000);
      json.clear();


      for(i=0;i<100;i++){
        arr.add(xReal[i]);
      }

    Serial.printf("Set array... %s\n", Firebase.RTDB.setArray(&fbdo, "fft
     arr.clear();

 for(i=100;i<200;i++){
        arr.add(xReal[i]);
      }

    Serial.printf("Set array... %s\n", Firebase.RTDB.setArray(&fbdo, "fft
 arr.clear();
```

```
 for(i=200;i<256;i++){
        arr.add(xReal[i]);
      }

    Serial.printf("Set array... %s\n", Firebase.RTDB.setArray(&fbdo,

        arr.clear();



  }

Serial.println("////////////");
delay(10000);

}
void PrintVector(double *vData, uint16_t bufferSize, uint8_t scaleTy
{
  for (uint16_t i = 0; i < bufferSize; i++)
  {
    double abscissa;
    /* Print abscissa value */
    switch (scaleType)
    {
      case SCL_INDEX:
        abscissa = (i * 1.0);
  break;
      case SCL_TIME:
        abscissa = ((i * 1.0) / samplingFrequency);
  break;
      case SCL_FREQUENCY:
        abscissa = ((i * 1.0 * samplingFrequency) / samples);
  break;
    }
    Serial.print(abscissa, 6);
    if(scaleType==SCL_FREQUENCY)
      Serial.print(" Hz");
    Serial.print("  ");
    Serial.println(vData[i], 4);
  }
```

```
  Serial.println();
}
void tostring(char str[], int num)
{
    int i, rem, len = 0, n;

    n = num;
    while (n != 0)
    {
        len++;
        n /= 10;
    }
    for (i = 0; i < len; i++)
    {
        rem = num % 10;
        num = num / 10;
        str[len - (i + 1)] = rem + ’0’;
    }
    str[len] = ’\0’;
}
```