Department of Electronic and Telecommunication Engineering

University of Moratuwa

Design Document

Industrial Machine Vibration Monitoring System

EN2160: Electronic Design Realization

Index Number	Name
210542B	R.M.L.H. Ratnayake
210549D	N.P.S.S. Rupasinghe

Group K (30)

Abstract

This documents focuses on the comprehensive details about the design process of a system for monitoring and analyzing vibrations of industrial machines.

As such, details on background research, component selection and PCB design, and enclosure design are included together with hardware, software, and firmware development for the design. It includes GUI application development, machine learning integration, and data visualization.

Moreover, in the appendices, more details are included in all of the above-mentioned aspects, as well as diaries of the team members, and design decisions taken at different points of time. Other than that, this document also includes a table containing the personal contributions of the group members to the project, and reviews from the other colleagues.

Contents

Abbreviations

Abbreviation	Definition
MCU	Micro Controller Unit
IMU	Inertial Measurement Unit
PCB	Printed Circuit Board
3D	3 - dimensional
DSP	Digital Signal Processing
FFT	Fast Fourier Transform
ML	Machine Learning
AI	Artificial Intelligence
I2C	Inter-Integrated Circuits (I^2C)
GUI	Graphical User Interface
PC	Personal Computer
IoT	Internet of Things
CAN	Controller Area Network
MQTT	Message Queuing Telemetry Transport
IC	Integrated Circuit
FTDI	Future Technology Devices International Limited
UART	Universal Asynchronous Receiver-Transmitter
LED	Light Emitting Diode
LSTM	Long Short-Term Memory
BOM	Bill Of Materials

Chapter 1

Background Research

1.1 Measuring Vibrations

There are many methods that we can measure the vibration of such devices. A few of them are provided below.

1. **Accelerometers**
    - Piezoelectric Accelerometers:
      These devices generate an electrical charge in response to mechanical stress. They are widely used for their broad frequency range and high sensitivity.
    - MEMS Accelerometers:
      Micro-Electro-Mechanical Systems (MEMS) accelerometers are compact and can be integrated into smart devices for continuous monitoring. They are suitable for high-volume applications due to their cost-effectiveness.
2. **Velocity Sensors**
    - Electromagnetic Velocity Sensors:
      These sensors use a coil and magnet system to measure velocity directly. They are robust and suitable for low to medium-frequency ranges.
    - Seismic Velocity Sensors:
      Often used in geophysical applications but can be adapted for industrial use to measure ground vibrations impacting machinery.
3. **Displacement Sensors**
    - Eddy Current Displacement Sensors:
      Utilize electromagnetic fields to measure the displacement of conductive materials. They are non-contact and suitable for monitoring rotating shafts and other critical components.
    - Capacitive Displacement Sensors:
      Measure changes in capacitance to determine displacement. They offer high precision and are ideal for small amplitude vibrations.
4. **Gyroscopes**
    - Measure the angular velocity and can be used in conjunction with accelerometers for comprehensive vibration analysis in multi-axis systems.

By considering all the requirements we decided to use an IMU which is an MPU6050 module to measure vibrations. which contains built-in accelerometers, Gyroscopes, and I2C communication.

1.2 Types of Vibrations

In industry, different varieties of industrial machines use different types of mechanisms to provide different needs required by people.
Mechanical vibrations in industrial machinery are a critical concern as they can lead to performance degradation, increased wear and tear, and ultimately, machine failure. Understanding the causes and mechanisms of these vibrations is essential for effective maintenance and design improvement.

we can discuss different types of industrial machine vibration types, concerning causes, mechanisms, and effect

1. **Rotational Imbalance**

   Cause:

   Rotational imbalance occurs when the center of mass of a rotating component, such as a rotor or flywheel, does not coincide with its axis of rotation.

   Mechanism:

   - The imbalance creates a centrifugal force that varies sinusoidally with the rotation speed. This force acts outward from the center of rotation and varies in magnitude as the component rotates.
   - As the rotational speed increases, the magnitude of the centrifugal force and, consequently, the amplitude of the vibration increases, typically proportional to the square of the speed.

   Effects:

   - Rotational imbalance can lead to excessive vibration, resulting in premature bearing failure, structural fatigue, and increased noise levels.
   - It is commonly addressed by balancing the rotating component, either statically or dynamically, to ensure the mass is evenly distributed around the axis of rotation.

2. **Misalignment**

   Cause:

   Misalignment occurs when two or more rotating shafts or components are not properly aligned along their intended axes. This can happen in both parallel and angular misalignments.

   Mechanism:

   - *Parallel Misalignment:* When the shafts are offset but parallel, causing them to exert alternating forces on each other as they rotate. This misalignment induces vibration at the rotational frequency of the shafts.
   - *Angular Misalignment:* When the shafts are at an angle to each other, resulting in periodic variations in angular velocity. This type of misalignment typically induces vibrations at twice the rotational frequency.

   Effects:

   - Misalignment generates additional forces on the bearings and couplings, leading to increased wear, overheating, and potential failure of these components.
   - It can be corrected by precise alignment of shafts using laser alignment tools or dial indicators.

3. **Eccentricity**

   Cause:

   Eccentricity arises when the geometric center of a rotating part, such as gears or pulleys, does not align with its rotational axis.

   Mechanism:

   - Eccentric components generate fluctuating loads on bearings and supports as they rotate, leading to periodic vibrations.
   - This condition often results from manufacturing defects, improper installation, or wear over time.

   Effects:

   - Eccentricity can cause oscillatory forces that lead to fatigue in machine components, resulting in cracks or fractures.
   - Regular inspection and precision manufacturing processes can minimize eccentricity while existing issues can be corrected by re-machining or replacing affected parts.

   With our selected components (MPU6050) we can measure anomalies related to all these vibrations.

Chapter 2

Our Product

As mentioned in the preliminary design document we have chosen our design from 4 main conceptual designs based on many criteria.

Refer to Preliminary Design Document
Chapter 4, Chapter 5, Chapter 6, Chapter 7

Briefly;
The product is based on **Fully-wired** and **Semi-modular** design and architecture.

The following chapters describe how we designed our product such that the above criteria are met.

Chapter 3

Component Selection, Schematics, and PCB Design

3.1 Component Selection

ATmega328P

As the main microcontroller of the product, we chose **ATmega328P-AU** SMD microcontroller by Atmel. It is a widely used microcontroller with a rich set of documentation and resources, as well as the following specifications[2] of it that go well with our requirements.

- 32KB flash program memory
- 2KB internal SRAM
- 16-bit timer counter with prescalar
- I2C and UART protocol support

FT232RL over CH340 series ICs

One of the main things we had to do in our PCB was to bridge USB ports to UART peripheral interface. In this case, it is a computer, and the ATmega328P microcontroller. This can be achieved using two main ICs available and widely used in the world.

1. FTDI FT232 series ICs
2. CH340 series ICs

However, since we got some bad ideas[7] about the CH340 chip such as the lack of market availability, we decided to use the IC by FTDI company, which is readily-available in the market, as well as many professional PCB designs worldwide had already used it and proved to be working well.

As a result, we chose **FT232RL by FTDI** USB-to-UART converter over the **CH340 series** ICs.

Other components

Most of the other components are selected to be SMD in the case size of 1206. The size is chosen such that handling and hand-soldering is not difficult.

Few exceptions like JST and screw terminal connectors, crystal oscillator, and headers are chosen to be through-hole, because of their availability.

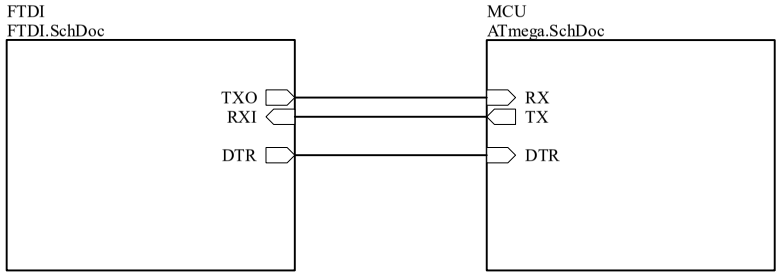
3.2 Schematic Design

Since the product includes a main PCB with the microcontroller, and a secondary unit (extendable) containing the IMU sensors, we had to design two separate PCBs for the two units.

1. Main controller schematic
   (a) Top-level schematic (hierarchy)
   (b) USB-to-UART converter schematic
   (c) MCU schematic
2. IMU sensor mount schematic

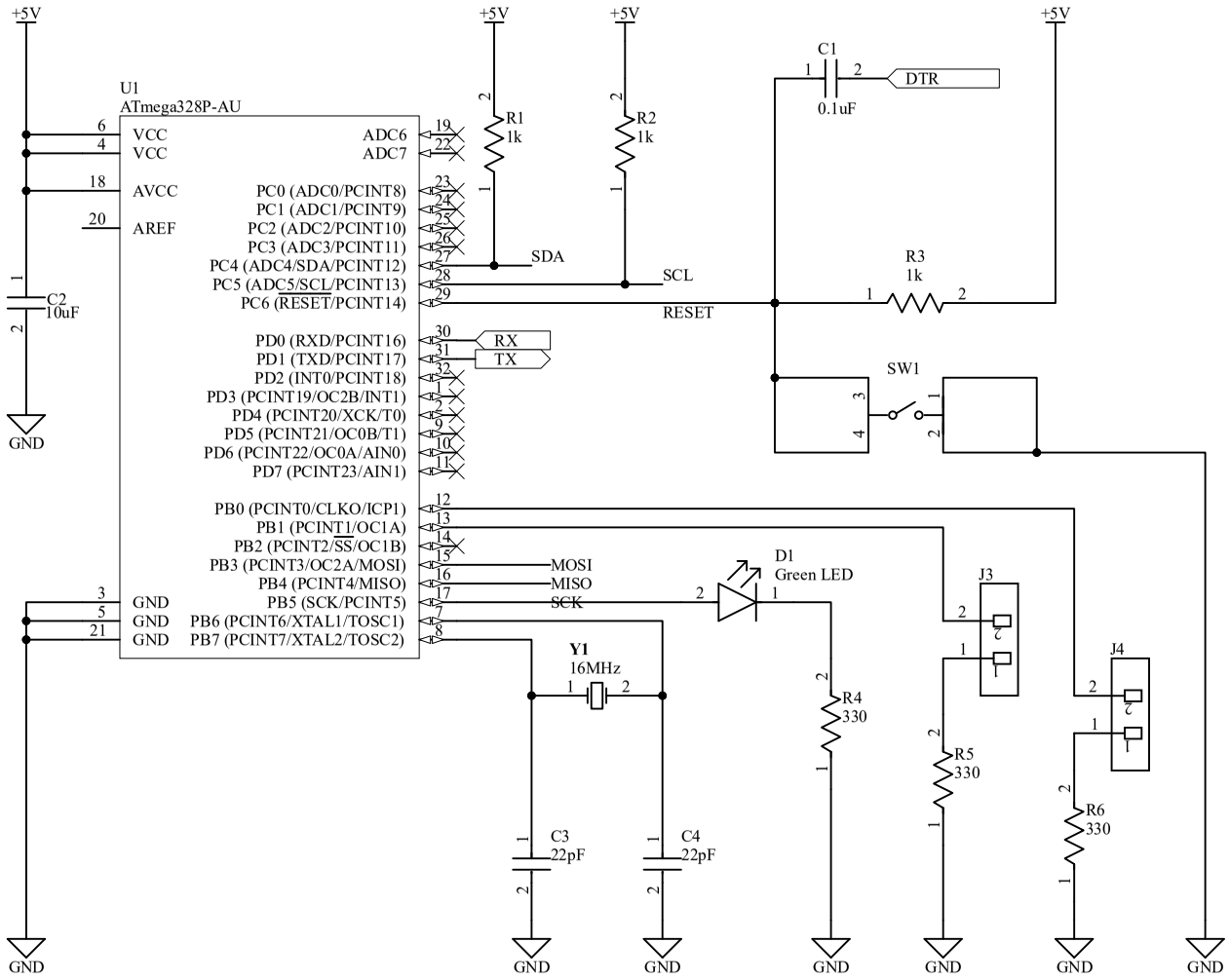
For the main PCB, hierarchical design was utilized, where we identified three main sub-parts. With the hierarchical design, it is possible to design schematics that are more readable, as well as flexible and easy to modify.

Schematics are attached below, in the order shown above.

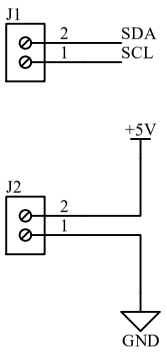


Title: Vibration Analyzer - Main		
Size: A4	Number: 1	Revision: v2.0
Date: 5-19-2024		Sheet of
File: D:\GitHub_Projects\..\Main.SchDoc		Drawn By: Linuka Ratnayake

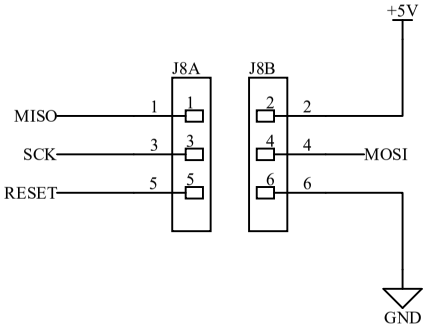
ATmega328P Microcontroller and Connections



Output Ports



Microcontoller - Burning Bootloader



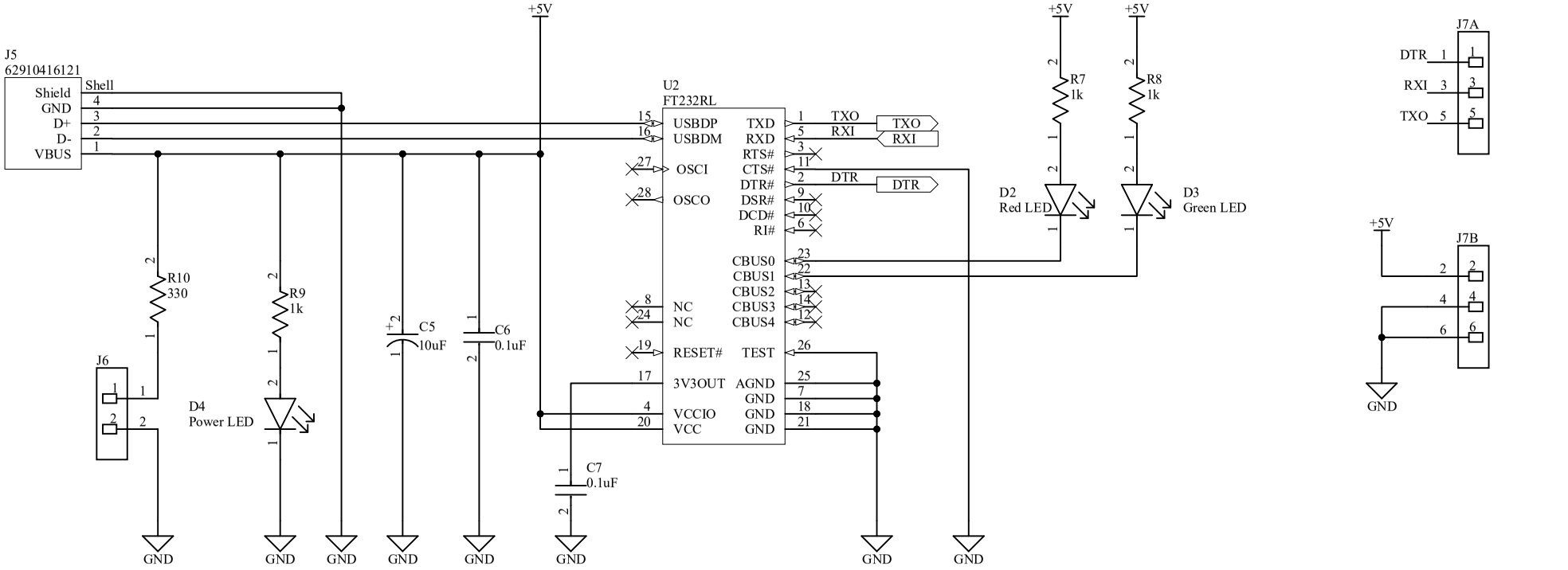
J8A, and J8B extends the SPI pins to faciliatate burning the bootloader to the microcontroller.

J3, and J4 pins can be used to give any kind of indications to the outside.

Title: Vibration Analyzer - MCU Circuit		
Size: A4	Number: 3	Revision: v2.0
Date: 5-19-2024		Sheet of
File: D:\GitHub_Projects\..\ATmega.SchDoc		Drawn By: Linuka Ratnayake

FTDI USB-to-UART Converter

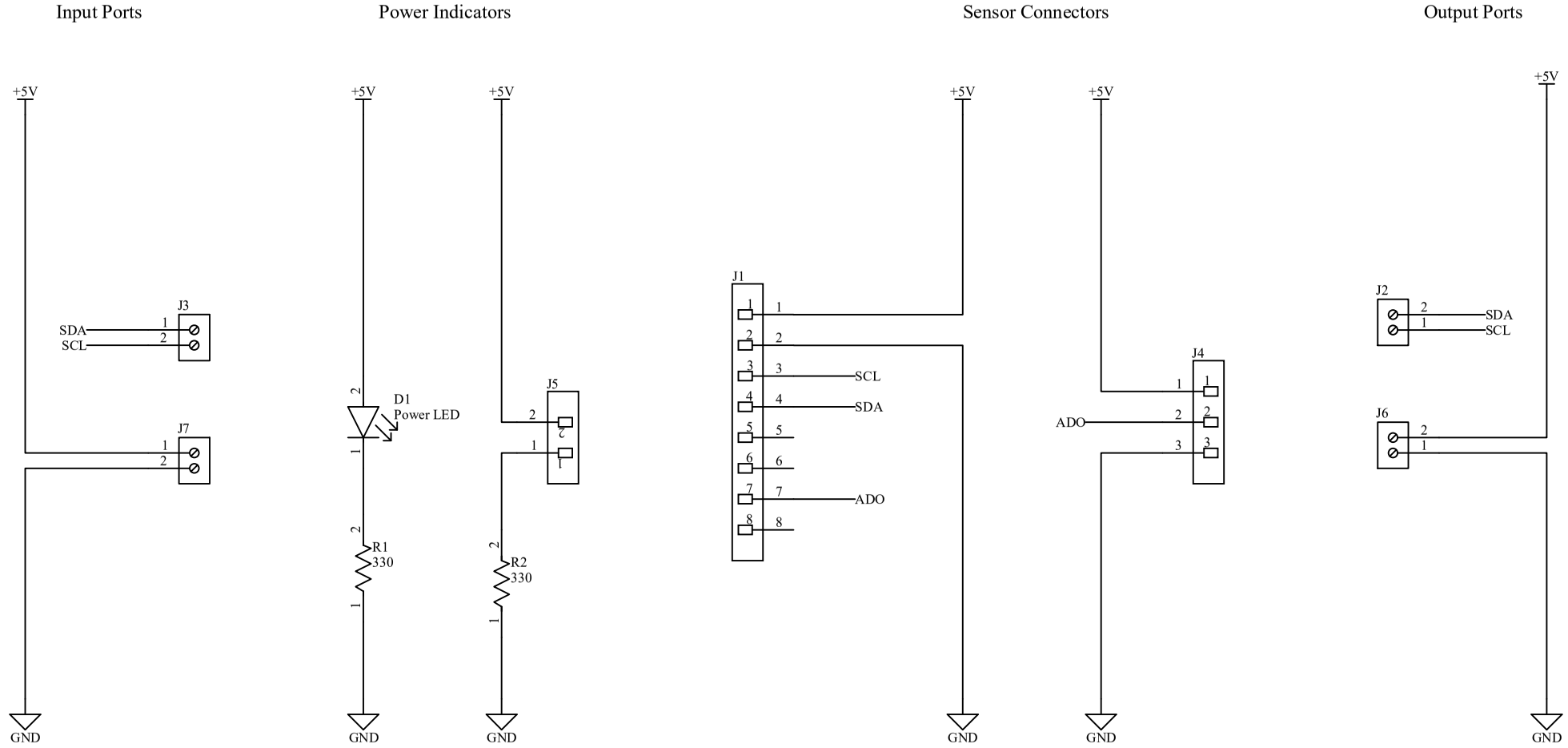
Microcontroller Programming



D4 - Internal, and J6 - External power indicators are used.

J7A, J7B - UART Rx and Tx are connected to external pins, for testing and as a bypass method of programming the microcontroller.

Title: Vibration Analyzer - FTDI Circuit		
Size: A4	Number: 2	Revision: v2.0
Date: 5-19-2024		Sheet of
File: D:\GitHub_Projects\..\FTDI.SchDoc		Drawn By: Sanuja Rupasinghe



One internal and one external power indicators are used.

Input and output ports are placed to facilitate cascading of multiple sensors.

With J1, IMU sensor can be directly soldered or connected using headers.

Title: Vibration Analyzer - Accelerometer Sensor Circuit		
Size: A4	Number: 4	Revision: v2.0
Date: 5-19-2024		Sheet of
File: D:\GitHub_Projects\..\Schematic.SchDoc		Drawn By: Linuka Ratnayake

3.3 PCB Design

PCB designing is carefully done considering many aspects.

- **Trace width**: Trace width is chosen to be 0.32mm for all the traces. Since there are no high-power components and hence high-power traces, this trace width is more than enough for power delivery and signals.
- **Use of via**: Vias of sufficient diameter are used to change the traces from the top plane to the bottom plane and vice versa.
- **Via stiching**: Via stiching connects the top and bottom ground planes for better signal integrity.
- **PCB Sizes**: The sizes are as follows.
  - Main Controller PCB - 70.1mm x 37.4mm
  - Sensor PCB - 46.6mm x 38.7mm
- **Component placement**: The placement of some components is crucial since they influence the signal quality and reduce noise. This especially includes the capacitors used for noise reduction in power lines.

The following pictures show different aspects of the PCB design process using software (not to scale).

1. Main Controller PCB Top View
2. Main Controller PCB Bottom View
3. Main Controller PCB 3D View
4. Sensor PCB Top View
5. Sensor PCB Bottom View
6. Sensor PCB 3D View

2D Views

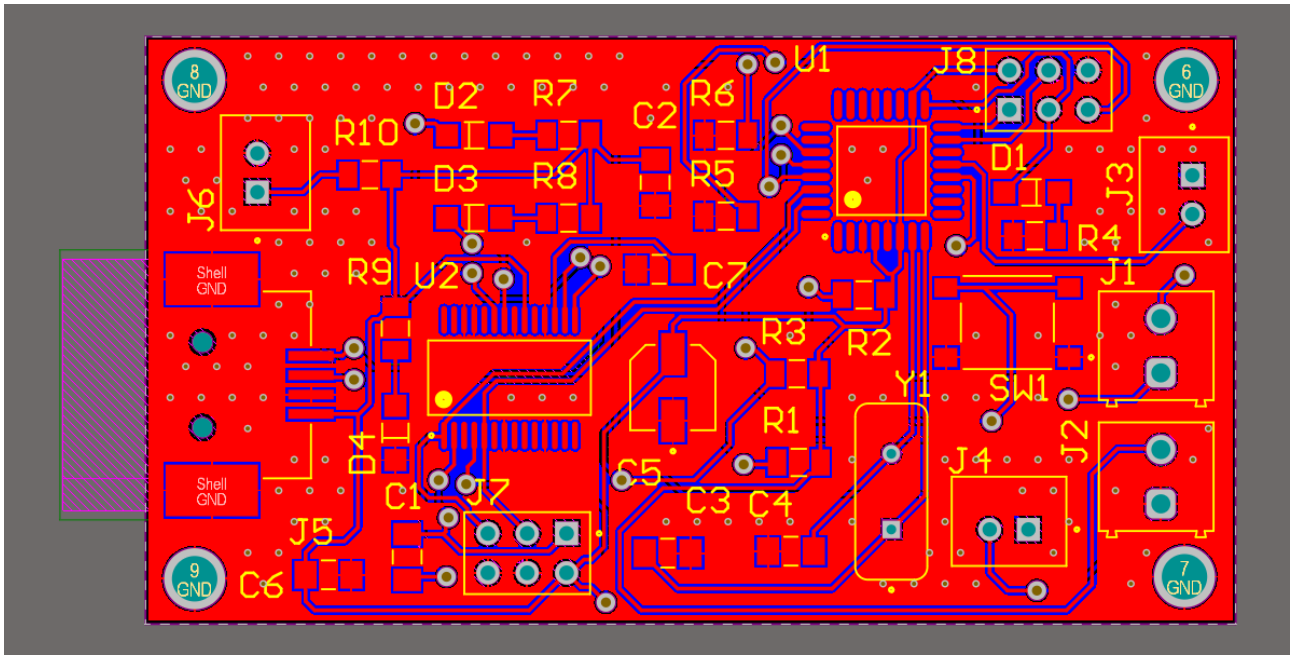


Figure 3.1: Main PCB (Top view)

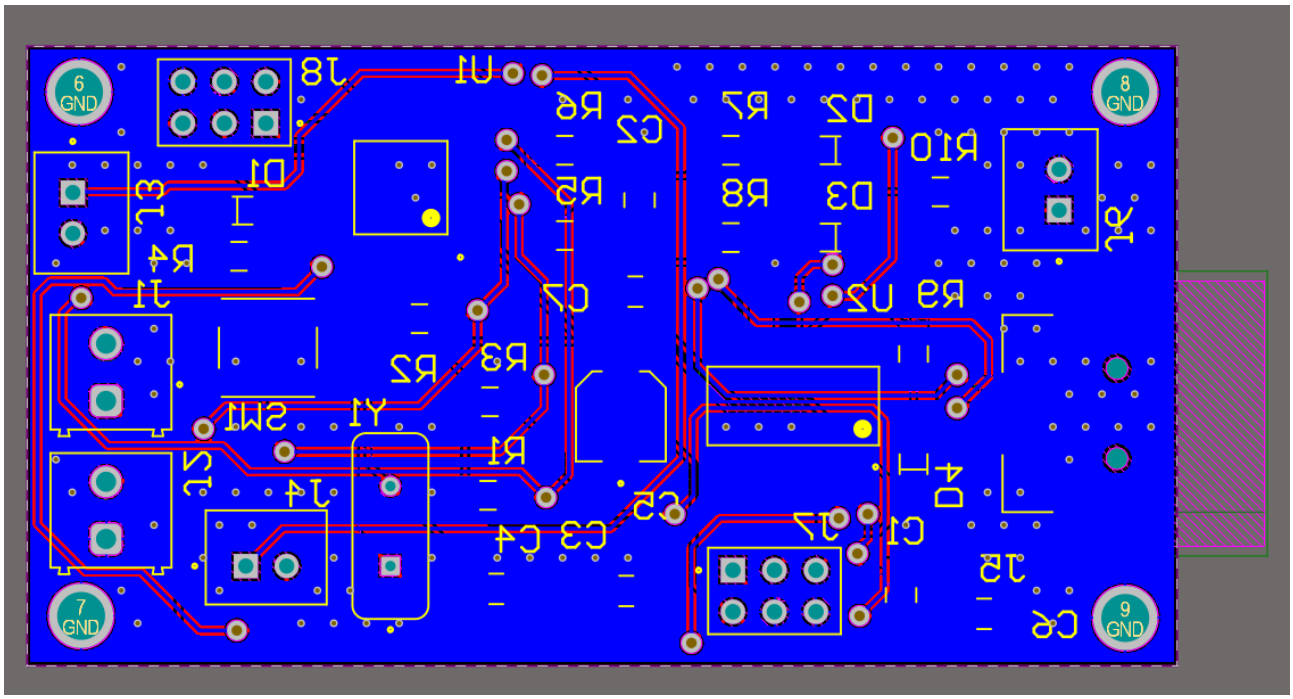


Figure 3.2: Main PCB (Bottom view)

3D View

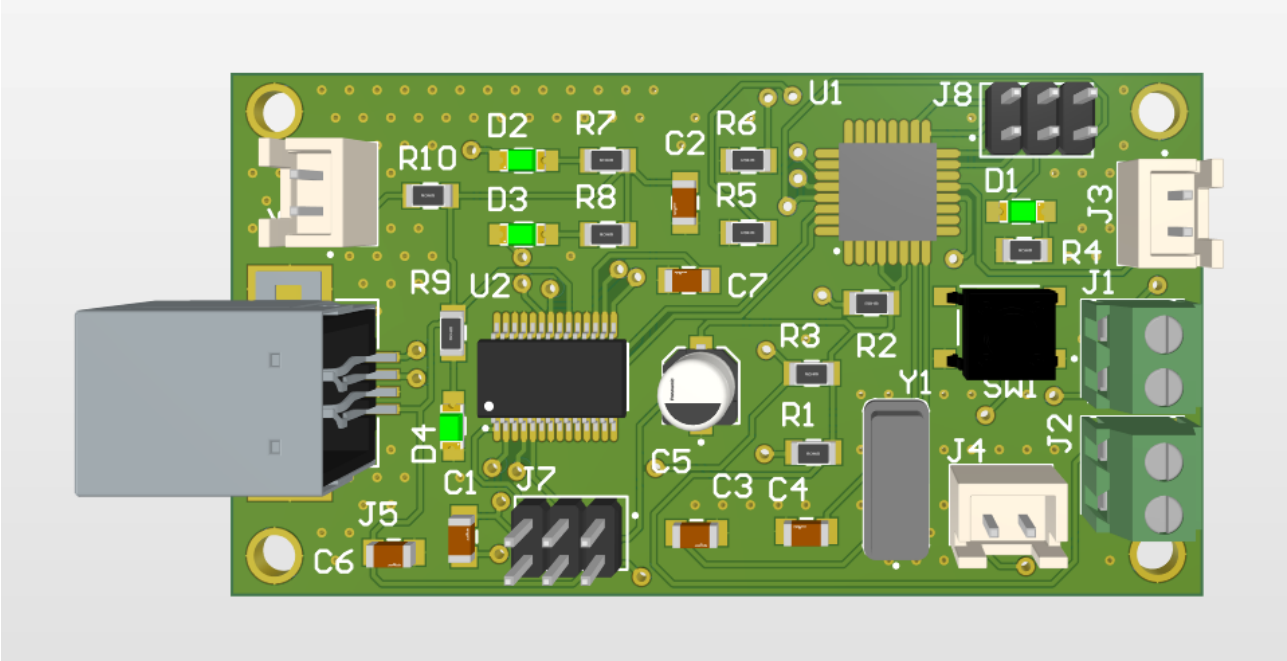


Figure 3.3: Main PCB (3D view)

2D Views

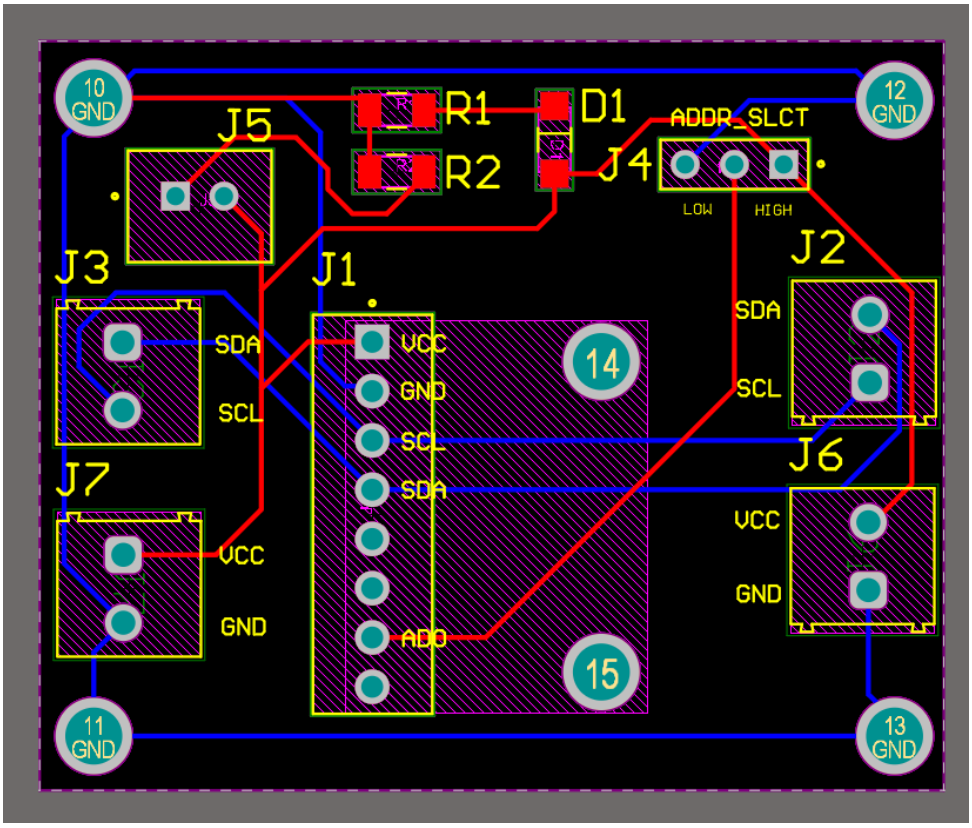


Figure 3.4: Sensor PCB (Top view)

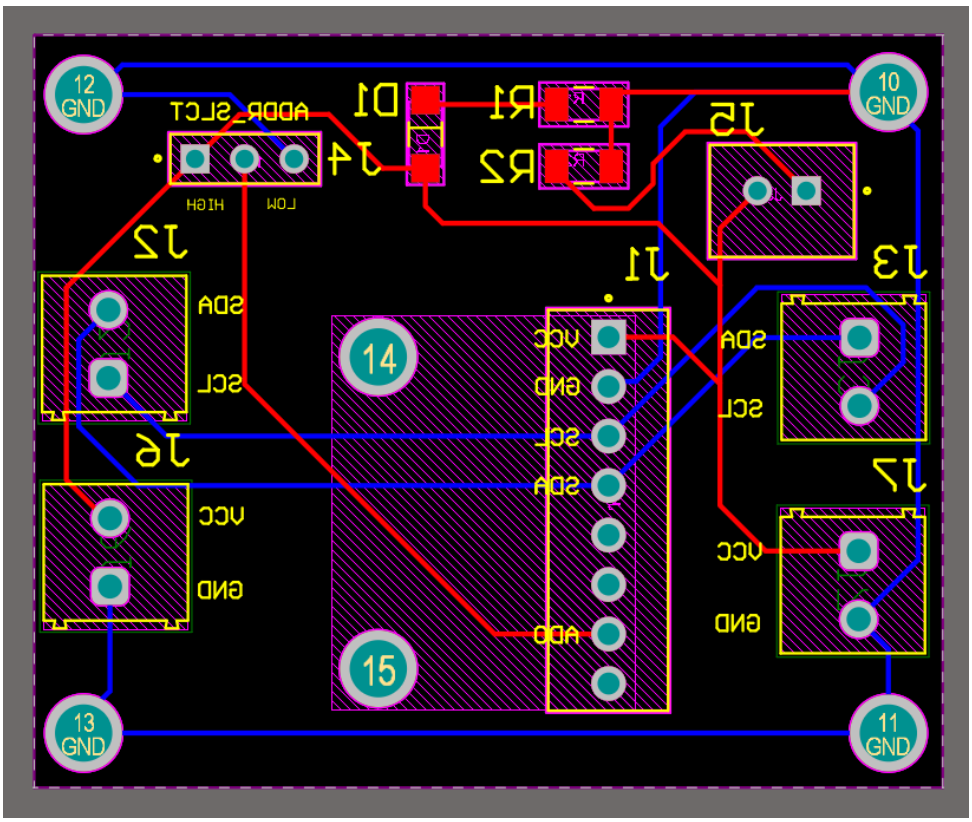


Figure 3.5: Sensor PCB (Bottom view)

3D View

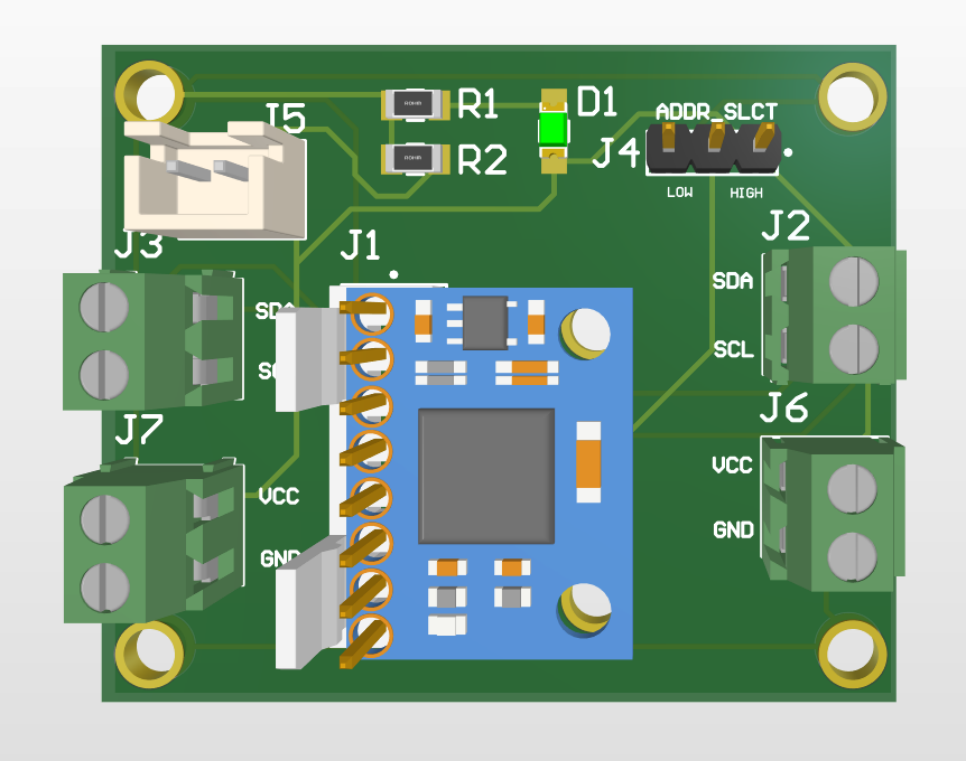


Figure 3.6: Sensor PCB (3D view)

The PCB Appendix contains:

1. Appendix C: PCB Photographs
    (a) *PCB photographs*
    (b) *Soldered PCB photographs*
    (c) *Working PCB photographs*
    (d) *PCB Sizes (Pictorial manner)*
2. Appendix D: BOM
    (a) *BOM for Main Controller*
    (b) *BOM for Module*

Chapter 4

Enclosure Design

The design of the enclosure for the industrial machine vibration monitoring system has been carefully developed to ensure durability, functionality, and aesthetic appeal. This report details the key aspects of the enclosure's design, materials, and functionality to ensure optimal performance in harsh industrial environments.

4.1 Overview and Specification

Enclosure Design Overview

1. Main Controller Enclosure:
    - This is a standalone unit designed to house the main controller, allowing for portability and flexibility in placement. Users can keep it anywhere in runtime.
2. Module Enclosures:
    - These are designed to house the individual modules and are intended to be attached directly to the vibration device.

The system's modular design enables easy integration and maintenance. The main controller and modules are housed separately to allow for independent placement and secure attachment to the vibration device.

Enclosure Design Specifications

1. Dimensions and Compactness
    - Main Controller Enclosure: Measures 10cm x 7cm, providing a compact and portable unit that can be easily placed near a laptop or other monitoring equipment.
    - Module Enclosures: Measure 9cm x 7cm, designed to be compact while ensuring all necessary components are securely housed.
2. Wall Thickness and Durability
    - The walls of the enclosures are 3mm thick, ensuring stability and durability in harsh vibrating environments. This thickness is optimal for long-term use, preventing deformation or damage due to continuous vibrations.
3. Color and Aesthetic

- Both enclosures are Grey, a color choice that minimizes the visual impact of dust and dirt in polluted environments. This ensures that the enclosures maintain a clean and professional appearance over time.

4. Weight
    - Main Controller Enclosure: Weighs 50g without any additional components, making it lightweight and easy to handle.
    - Module Enclosures: Weigh 70g without any additional components, balancing durability with ease of installation.
5. Material
    - The enclosures are made from PLA+ (Polylactic Acid Plus), a material known for its strength, durability, and ease of molding. PLA+ is chosen for its enhanced properties over standard PLA, offering better heat resistance and mechanical strength.
6. 3 mm screws
    - We are using 4, 3 mm screws to connect lids to it's bases in this case connect Main Controller Lid to Main Controller Base and connect Module Lid to Module Base.

Moldability

- Both the main controller and module enclosures are designed to be moldable, facilitating mass production and ensuring consistency in quality. We have put 1-degree draft angle.

4.2 Connectoin and Wiring

Connectors

we are using standard industrial connectors to connect our main controller and modules.

- We are using standard industrial cable M12 connector 4-pin aviation plugs to establish electrical connections between the modules and the main controller.
- These connectors provide a robust mechanical connection, crucial for maintaining stability and reliability in high-vibration environments.
- The M12 connectors are IP67 rated, ensuring protection against dust and water ingress, which is essential for maintaining performance in harsh industrial conditions.
- The connectors feature a screw-lock mechanism that prevents accidental disconnection, enhancing the overall safety and reliability of the system.
- Easy to install and replace, the M12 connectors contribute to the modularity and ease of maintenance of the entire system.

To learn more about the M12 connector 4-pin aviation plug visit [12]

Wiring Convension

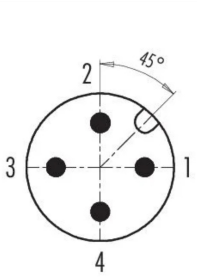


Figure 4.1: Wiring Diagram

1. 5V 2. GND 3. SCL 4. SDA

M-12 4-Pin Connector

The following pages show the Design of the Enclosure and the Architecture of the M12 4-pin connector. For more information about dimensions and details visit the website [13]

Figure 4.2: physical appearance

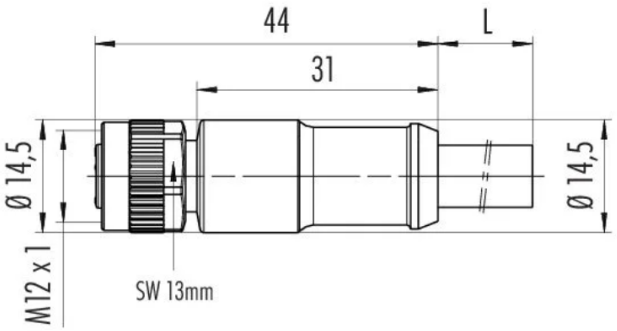


Figure 4.3: Dimensions

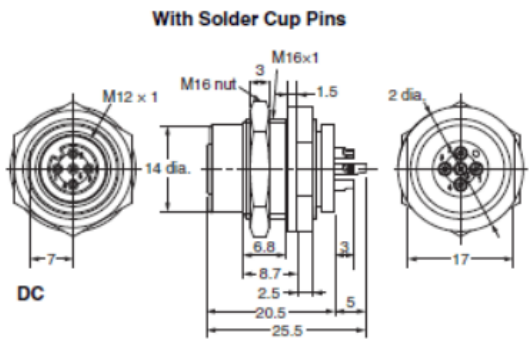


Figure 4.4: physical appearance

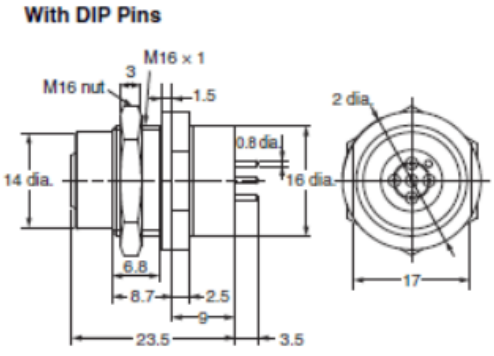
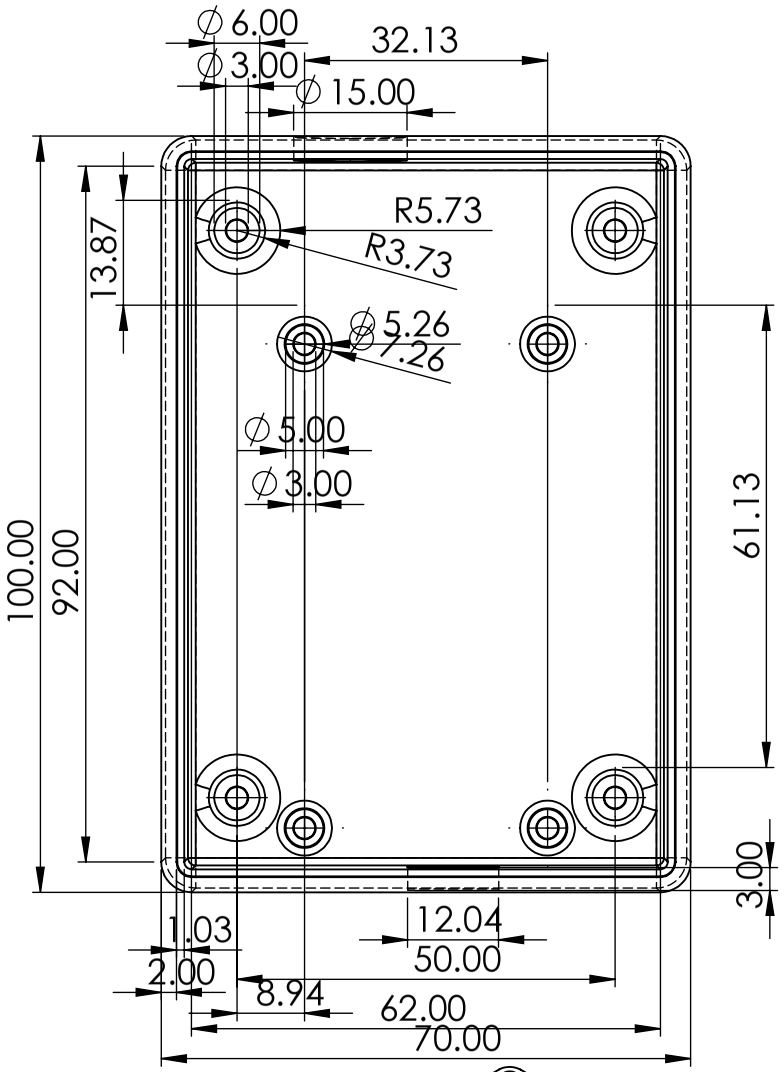
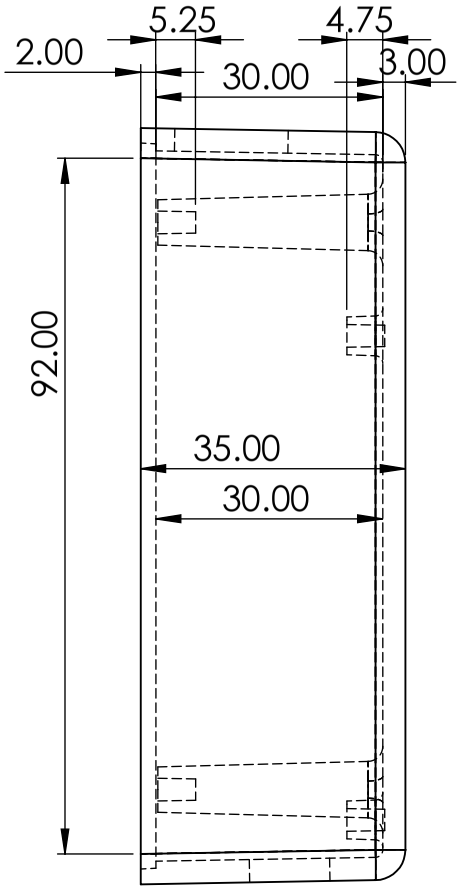
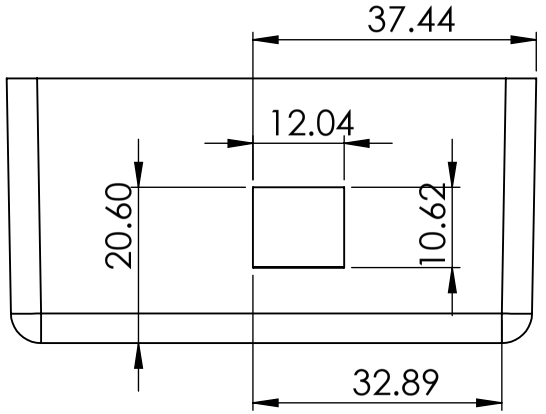
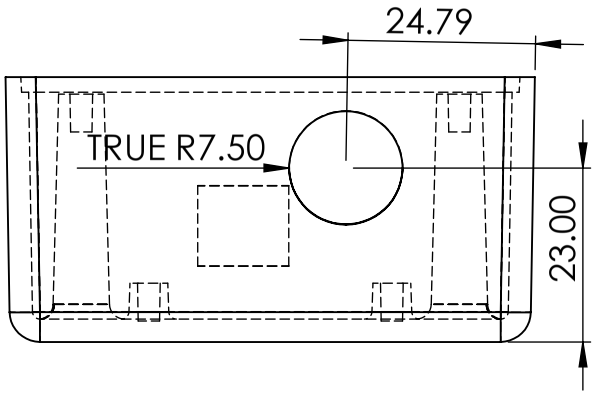


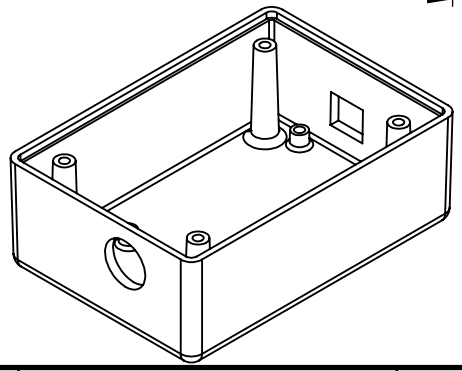
Figure 4.5: Dimensions



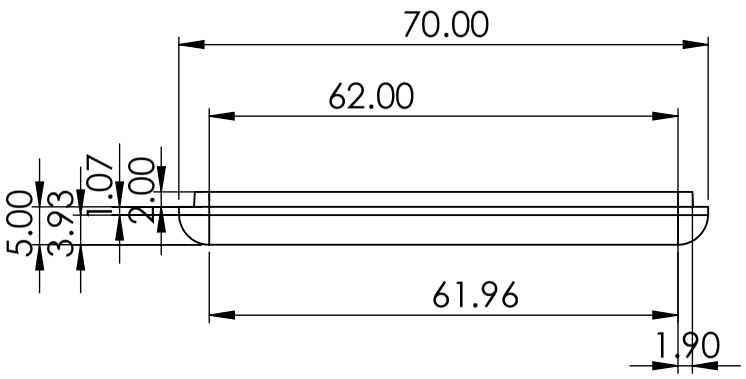


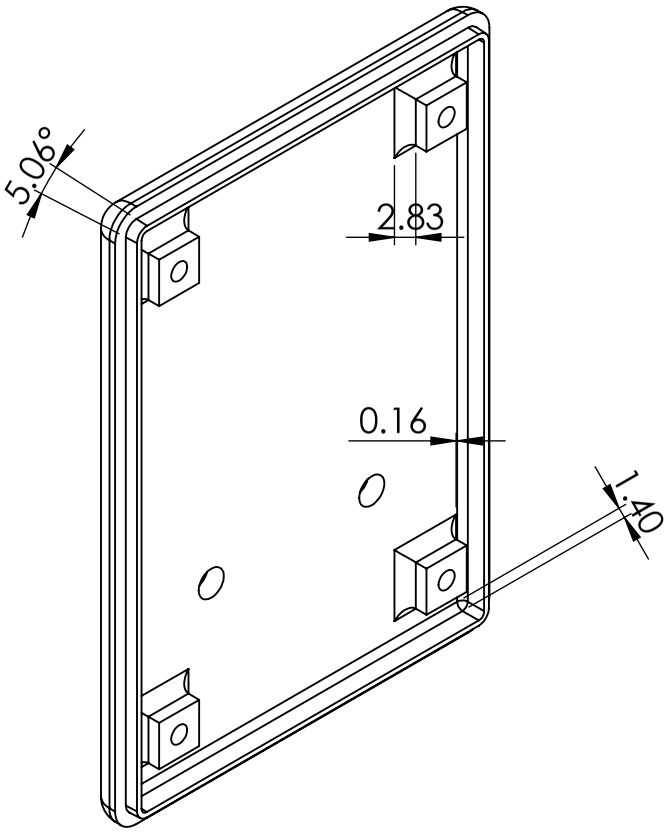


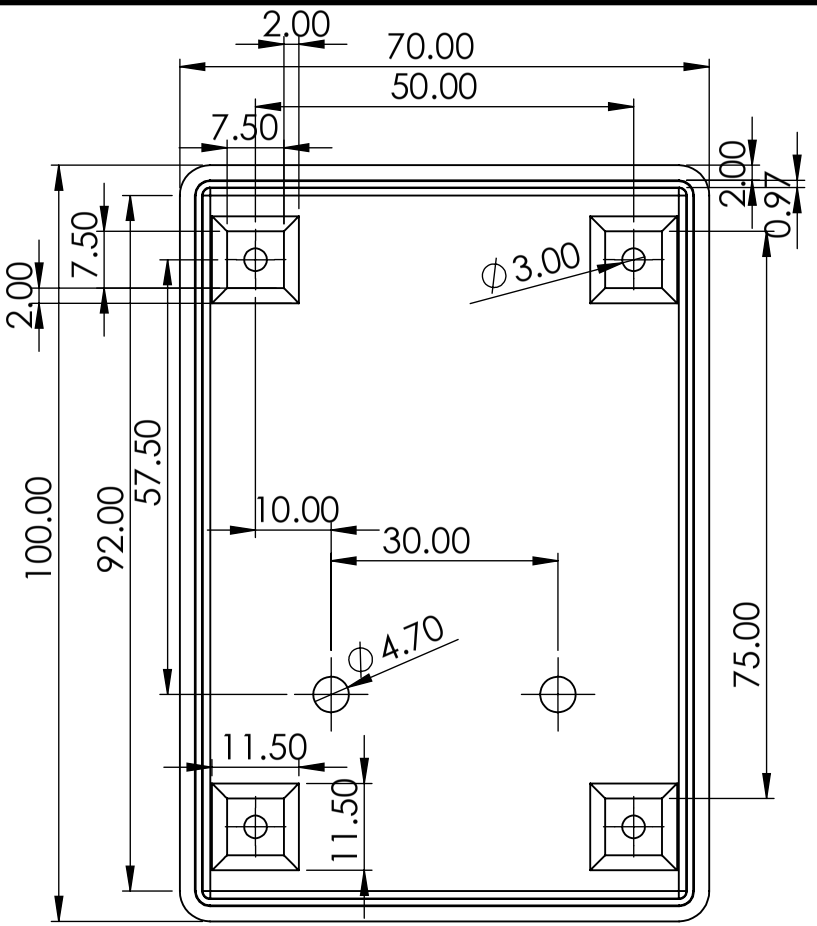


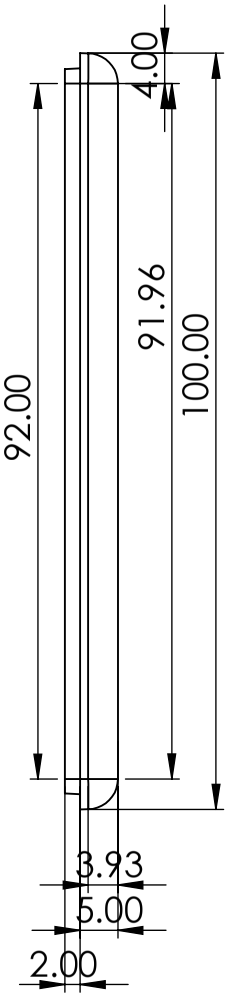


UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN GIVEN MEASUREMENT UNNITS. SURFACE FINISH: LINEAR: mm ANGULAR: degrees			Date 14/05/2024	DO NOT SCALE DRAWING	REVISION Rev_1
			Company/University University of Moratuwa	TITLE: Main_Controller_Base	
	NAME	SIGNATURE	DATE		
DRAWN	Sanuja Rupasinghe		14/05	DWG NO. 01	A4
CHK'D					
APPV'D					
			WEIGHT:	SCALE:1:2	SHEET 1 OF 1

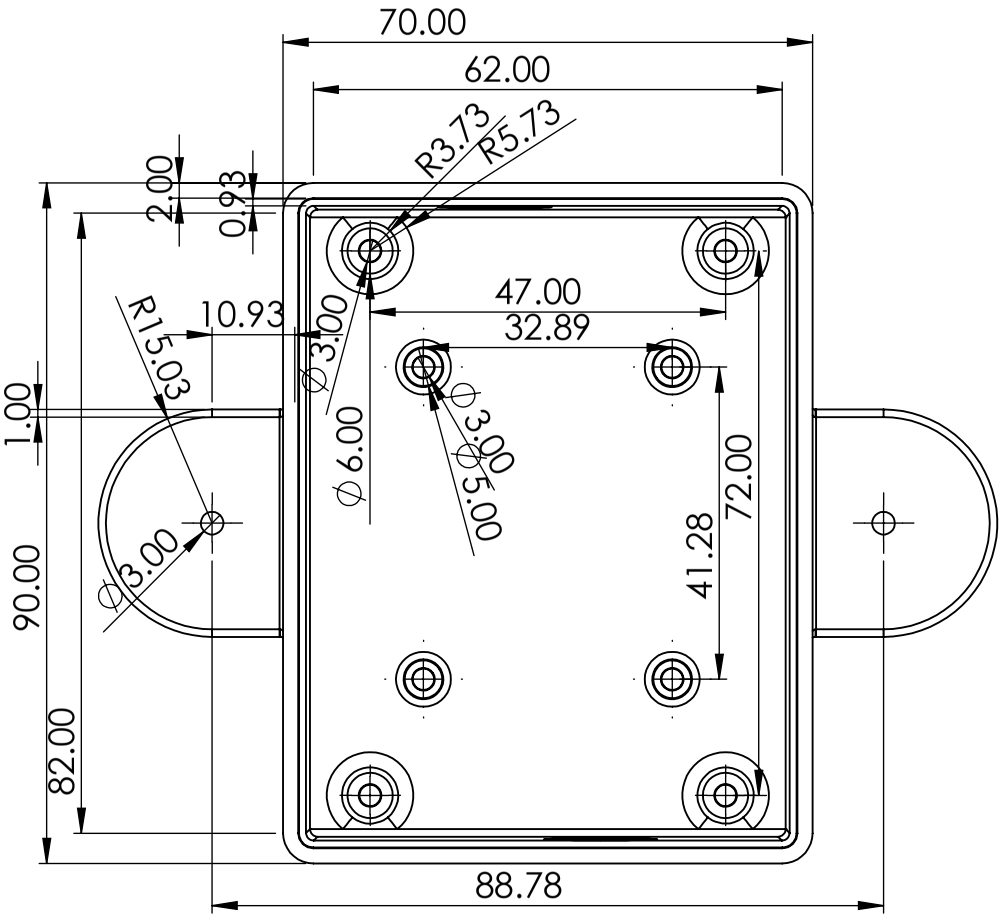


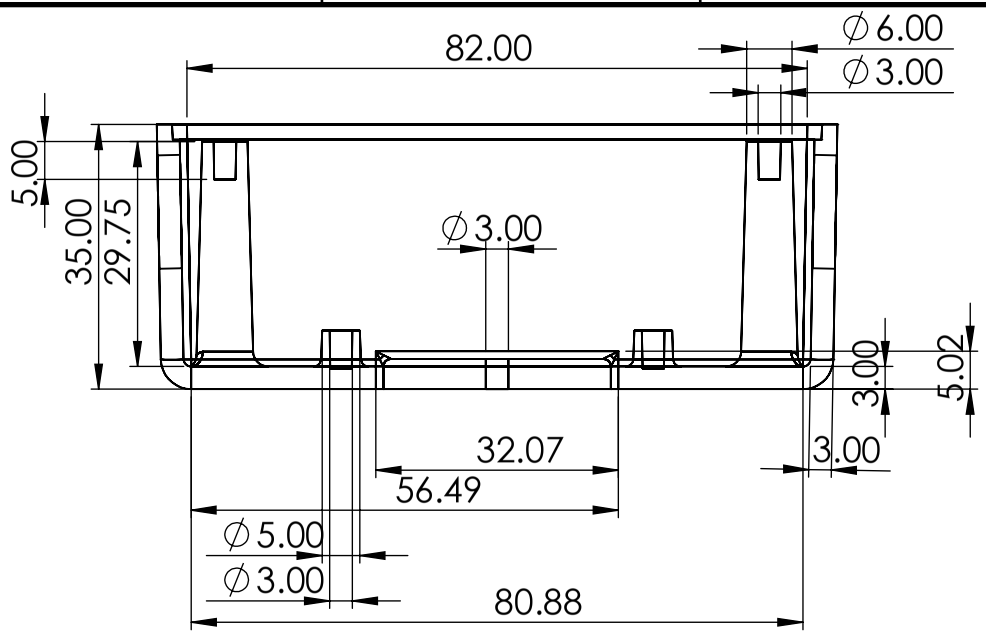


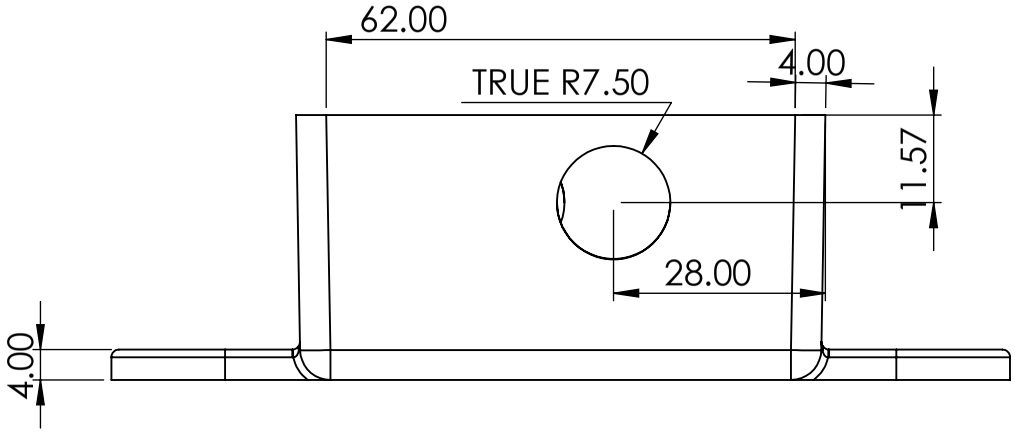


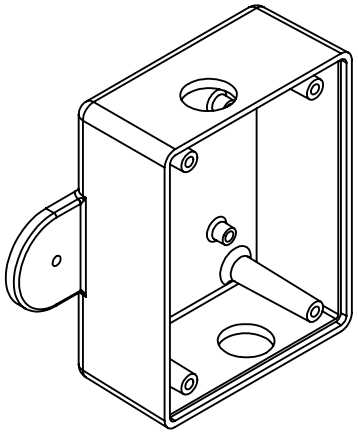


UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN GIVEN MEASUREMENT UNNITS. SURFACE FINISH: LINEAR: mm ANGULAR: degrees			Date 14/05/2024	DO NOT SCALE DRAWING	REVISION Rev_1
			Company/University University of Moratuwa	TITLE: Main_Controller_Lid	
	NAME	SIGNATURE	DATE		
DRAWN	Sanuja Rupasinghe		14/05	DWG NO. 01	A4
CHK'D					
APPV'D					
			WEIGHT:	SCALE:1:2	SHEET 1 OF 1

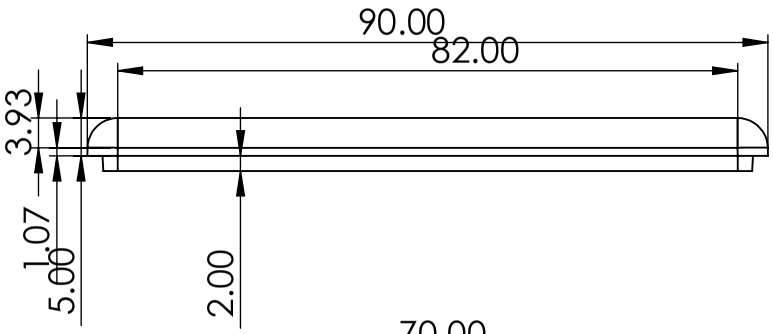


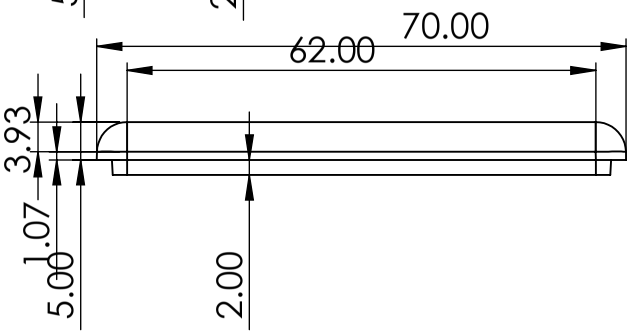


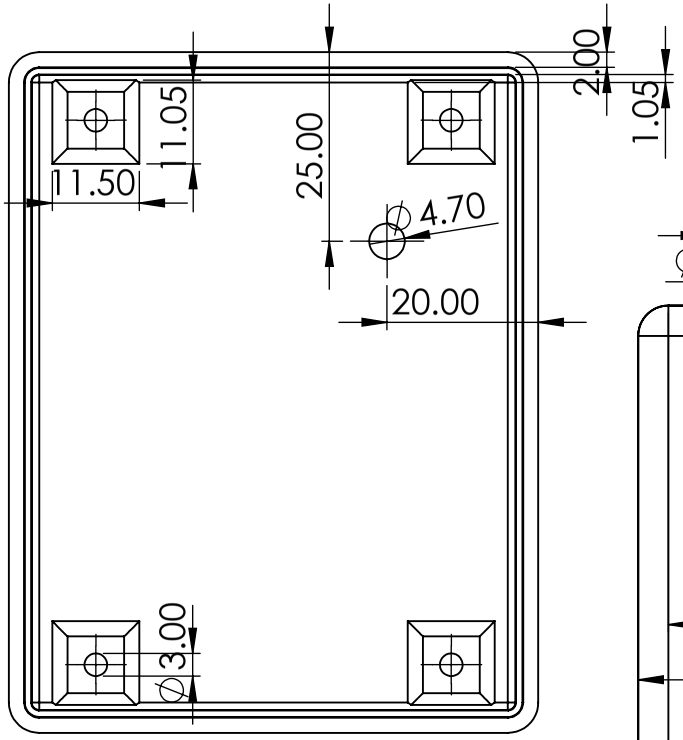


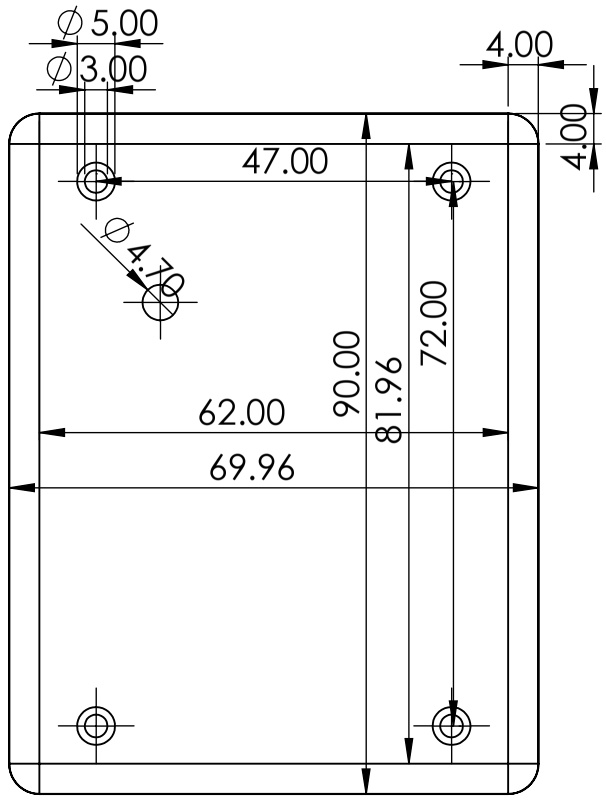


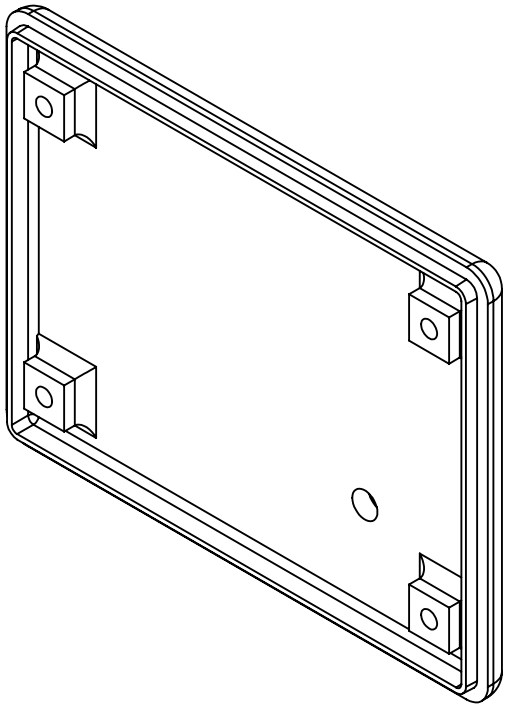
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN GIVEN MEASUREMENT UNNITS. SURFACE FINISH: LINEAR: mm ANGULAR: degrees			Date 14/05/2024	DO NOT SCALE DRAWING / REVISION Rev_1
			Company/University University of Moratuwa	TITLE: Module_Base
	NAME	SIGNATURE	DATE	
DRAWN	Linuka Ratnayake		14/05	DWG NO. 01 / A4
CHK'D				
APPV'D				
			WEIGHT:	SCALE:1:2 / SHEET 1 OF 1











UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN GIVEN MEASUREMENT UNNITS. SURFACE FINISH: LINEAR: mm ANGULAR: degrees			Date 14/05/2024	DO NOT SCALE DRAWING	REVISION Rev_1
			Company/University University of Moratuwa	TITLE: Module_Lid	
	NAME	SIGNATURE	DATE		
DRAWN	Linuka Ratnayake		14/05	DWG NO. 01	A4
CHK'D					
APPV'D					
			WEIGHT:	SCALE:1:2	SHEET 1 OF 1

4.3 3D model of enclosure designs

Main Controller

Main Controller Base

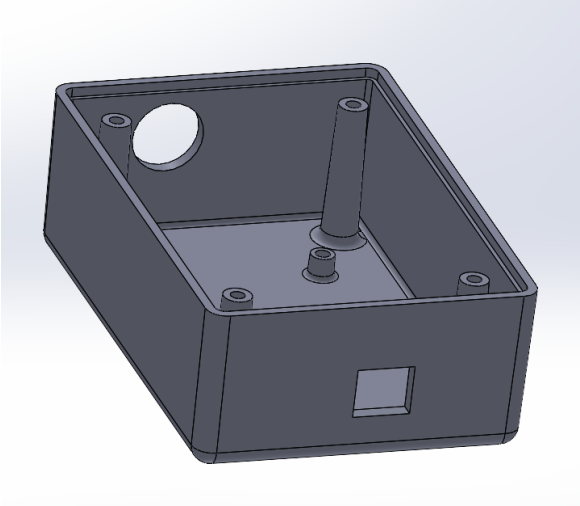


Figure 4.6: View 1

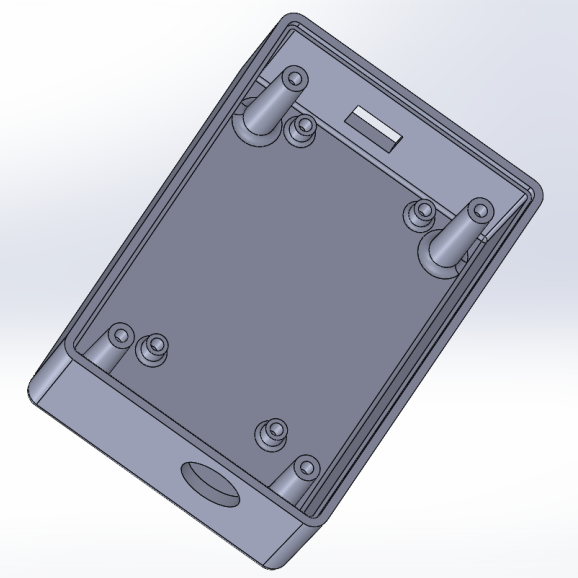


Figure 4.7: View 2

Main controller Lid

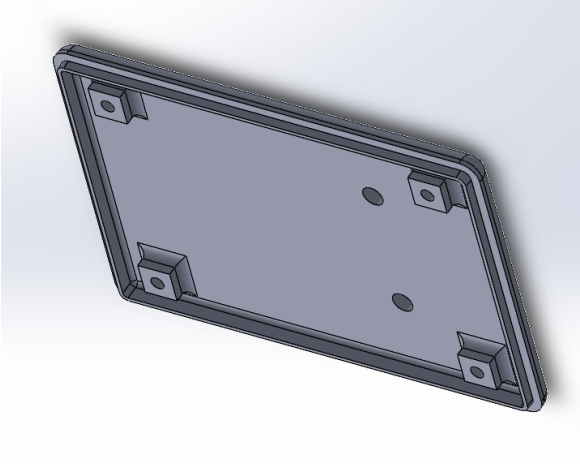


Figure 4.8: View 1

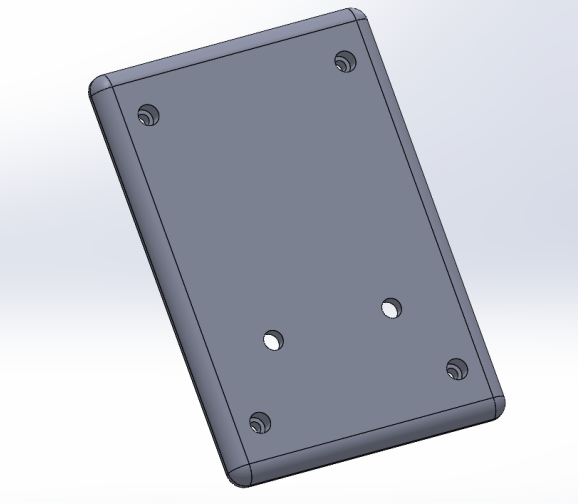


Figure 4.9: View 2

Module

Module Base

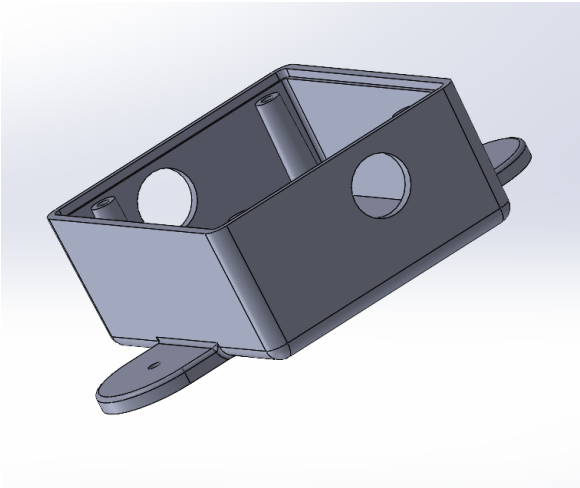


Figure 4.10: View 1

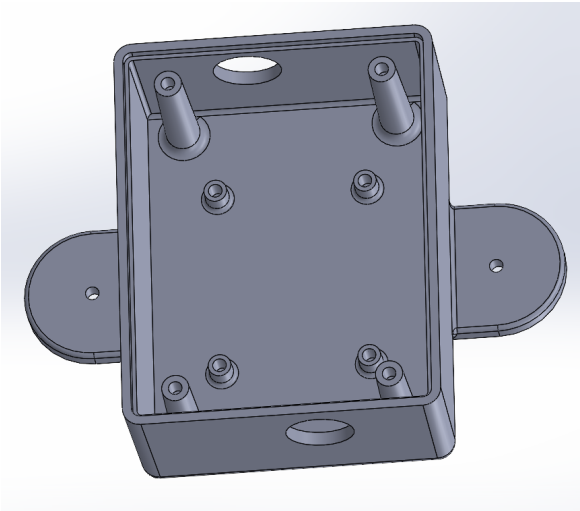


Figure 4.11: View 2

Module Lid

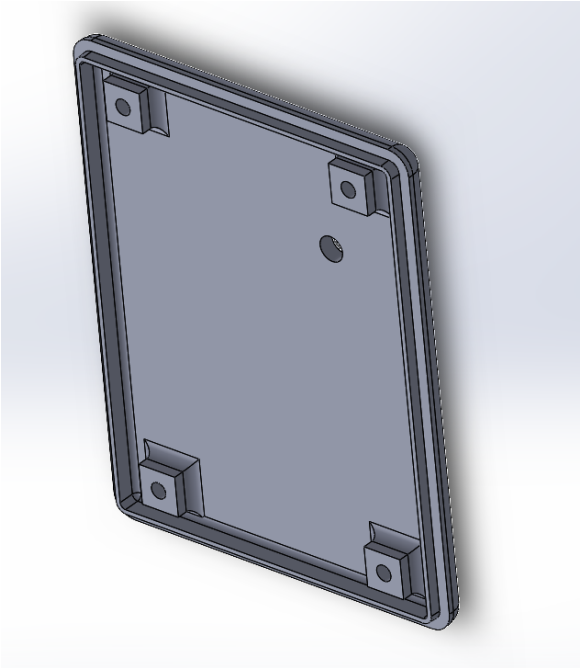


Figure 4.12: View 1

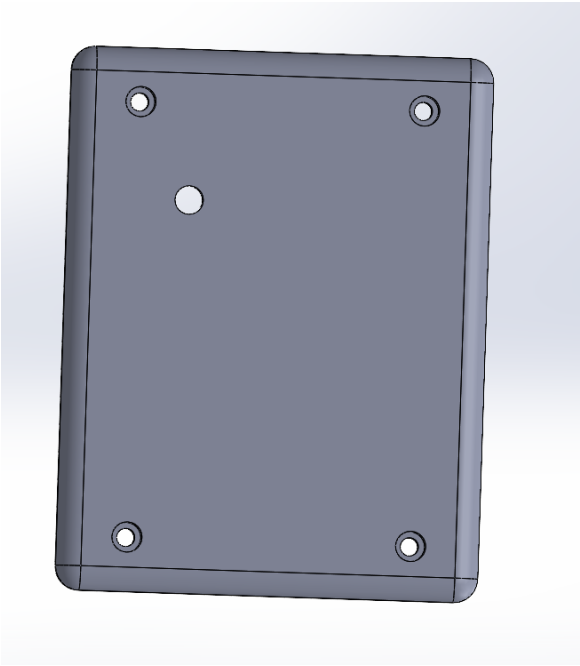


Figure 4.13: View 2

Assemblies

Assemblies Without Components

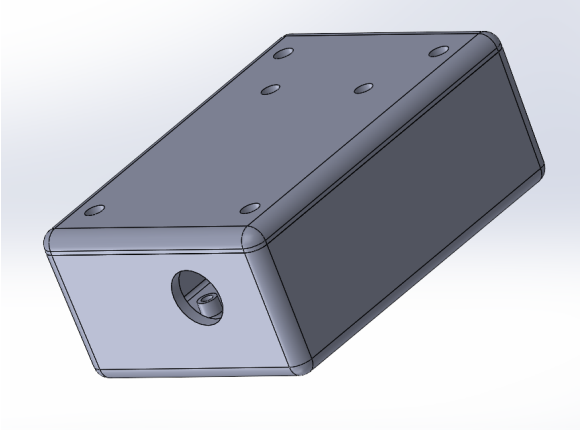


Figure 4.14: Main Controller Assembly

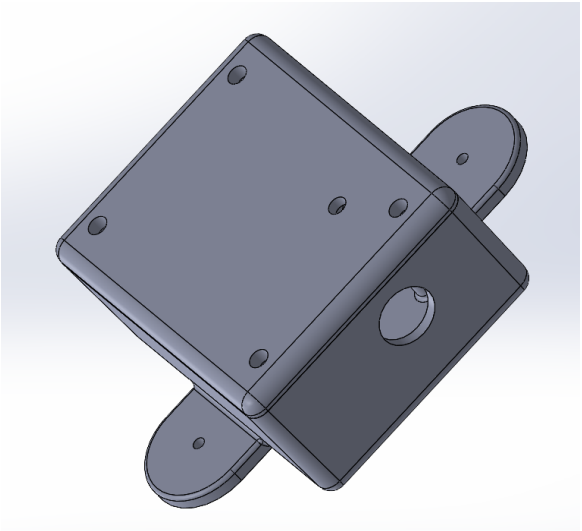


Figure 4.15: Module Assembly

Assemblies With Components

Main Controller

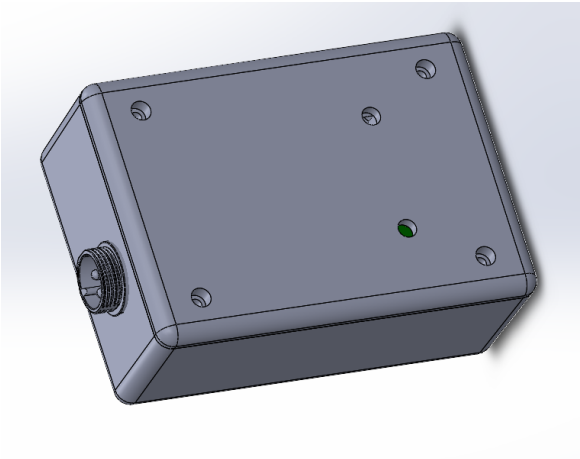


Figure 4.16: Main Controller Assembly 1

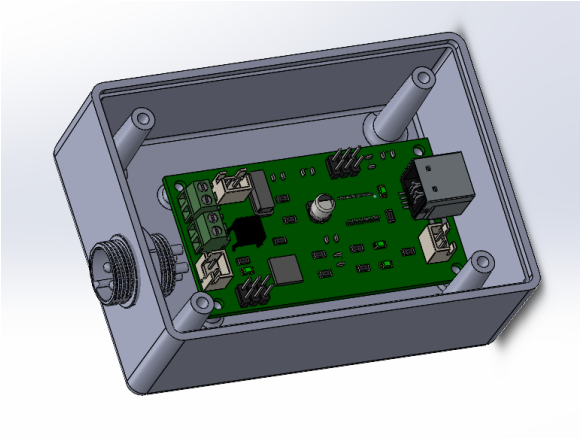


Figure 4.17: Main Controller Assembly 2

Module

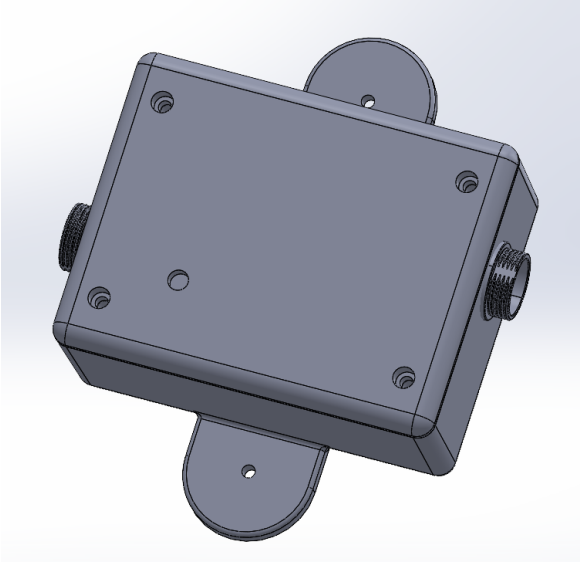


Figure 4.18: Module Assembly 1

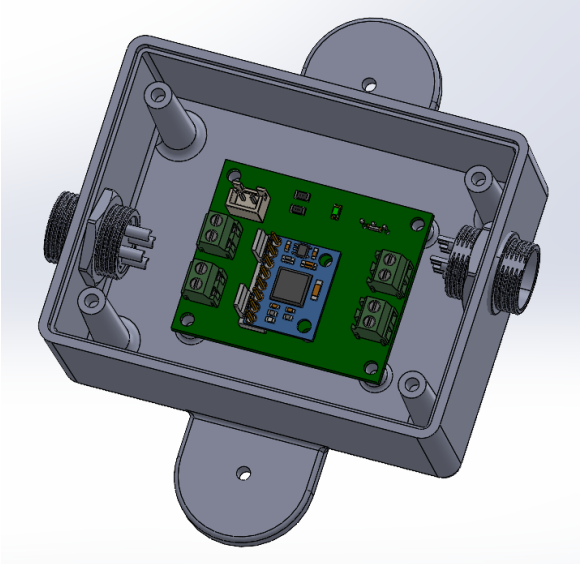


Figure 4.19: Module Assembly 2

The Enclosure Appendix contains

1. *Solid Works Hierarchy*
2. *Draft Analysis*
3. *Pictures Of printed enclosure (without Components)*
4. *Pictures Of printed enclosure (with Components)*

Chapter 5

Microcontroller Unit and Sensor(s)

5.1 Implementation and Data Flow

As depicted in the following figure, the entire system can be divided into three main parts: the microcontroller unit, sensor(s), and the computer where the user software runs.

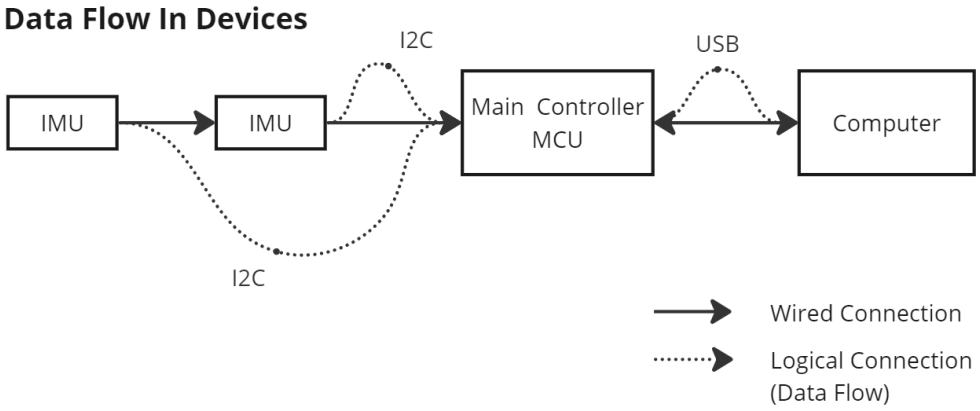


Figure 5.1: Wiring Convension

The microcontroller unit is responsible for the following tasks.

- Connecting with one or more sensor units via I2C protocol.
- Sample data from accelerometer sensors at a given frequency, and store them inside until the buffers of a given size are full.
- Once the buffers are full, send data to the computer via UART protocol.
- Listen to the computer for any alerts, and if received turn the alert mechanism on. (e.g. blinking an LED)

The above behavior is facilitated by the main microcontroller: **ATmega328P-AU**, which is carefully coded and tested using C++. Other than that the microcontroller unit consists an FTDI USB-to-UART converter IC to facilitate communication with the computer.

5.2 Sampling Frequency and Timer Interrupts

ATmega328P works on a 16MHz clock and has an internal counter of 16 bits, which can store a maximum of $2^{16} = 65536$. Hence, according to the way the timer interrupts are utilized in the program it gives an interrupt each time the counter register overflows.

Clock frequency

$$f_{clk} = 16\,\text{MHz}$$

Buffer size

$$\begin{aligned} x &= 2^{16} \\ &= 65536 \end{aligned}$$

Prescalar value

$$p = 1, 8, 64, 256, 1024$$

Sampling frequency

$$\begin{aligned} &= f \\ &= \frac{1}{T} \end{aligned}$$

where T is the sampling time period.

$$\frac{f_{clk}}{p} \times T = (x - x_0)$$

$$\frac{f_{clk}}{p} \times \frac{1}{f} = (x - x_0) \tag{5.1}$$

where timer buffer starting value is x_0.

Here, in our design we sample at 200 Hz, and prescalar is chosen to be $p = 8$. We can find the value for x_0.

From 5.1,

$$\frac{16 \times 10^6}{8} \times \frac{1}{200} = (2^{16} - x_0)$$

$$x_0 = 55536$$

Actually, the microcontroller is coded in such a way that once the desired sampling frequency is set via a variable, it automatically selects the suitable pre scalar value, and the counter starting value using the above 5.1 equation.

Chapter 6

Machine Learning for Anomaly Detection

6.1 Overall ML Model Architecture

We developed a sequential model in **Tensorflow** using LSTM layers to work as an **autoencoder**. The model summary is as follows.

Model: "sequential"

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 30, 128)	66,560
lstm_1 (LSTM)	(None, 64)	49,408
repeat_vector (RepeatVector)	(None, 30, 64)	0
lstm_2 (LSTM)	(None, 30, 64)	33,024
lstm_3 (LSTM)	(None, 30, 128)	98,816
time_distributed (TimeDistributed)	(None, 30, 1)	129

Total params: 247,937 (968.50 KB)
Trainable params: 247,937 (968.50 KB)
Non-trainable params: 0 (0.00 B)

Figure 6.1: Model summary

It should be noted that we utilize $3n$ identical copies of this model architecture for the n separate IMU sensors, which send data for the x, y, and z axes separately. For the ease of understanding, suppose we use only one IMU sensor ($n = 1$), and hence three ($3 \times n = 3 \times 1 = 3$) ML models.

6.2 Model Behavior

The models are first created untrained. The user is intended to collect some test data and train the models before attempting to predict anomalies. Refer to chapter 7 for detailed info on different ways to train the model.
Once trained, the models will continuously inference incoming data points to detect any anomalies.

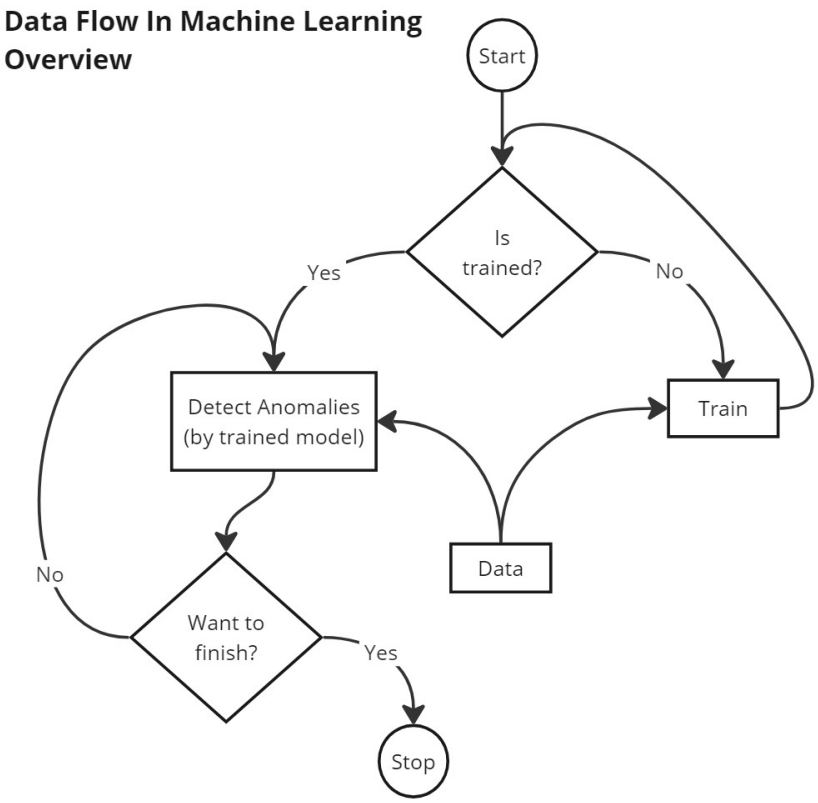


Figure 6.2: ML model behavior

6.3 Integration of ML Models to Detect Anomalies

The MCU is responsible for collecting and sending accelerometer data as batches of predefined size. Once those data are received by the computer, they are sent through the individual ML models (inferencing).

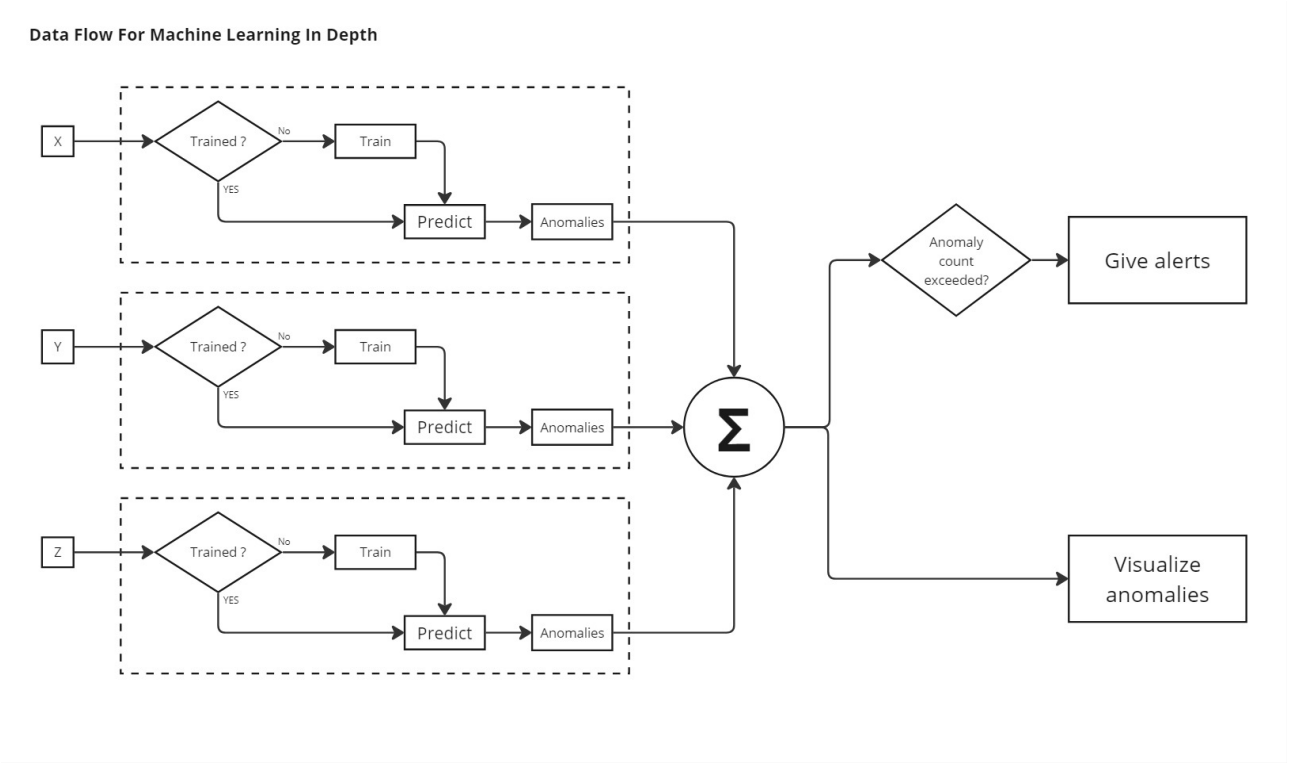


Figure 6.3: ML Model Integration

The models identify and generate a list of anomaly indices present in their respective datastream/ axis. They will be concatenated and checked against a given threshold for anomaly indices count. This is important since it is not the best approach to label the machine is behaving anomalously once a single anomaly point is detected. So, the threshold value plays an important role in adjusting the sensitivity.

Finally, if the anomaly count exceeds the predefined threshold, alerts will be activated, all the while anomalies are visualized graphically on the screen.

6.4 Alert Mechanism

The mechanism for giving alerts is analogous to the mechanism in a Schmidt trigger.

When the number of anomalies present exceeds the threshold for the first time, it starts a timer (say *timer1*). Moreover, every time the number of anomalies present exceeds the threshold, it resets another timer (say *timer2*).
The system starts to give alerts when the time elapsed since the first anomaly detection exceeds some predefined value, and the time elapsed since the last occurrence is less than some other predefined value.
If either of those conditions fails to satisfy, it goes back to its original state.

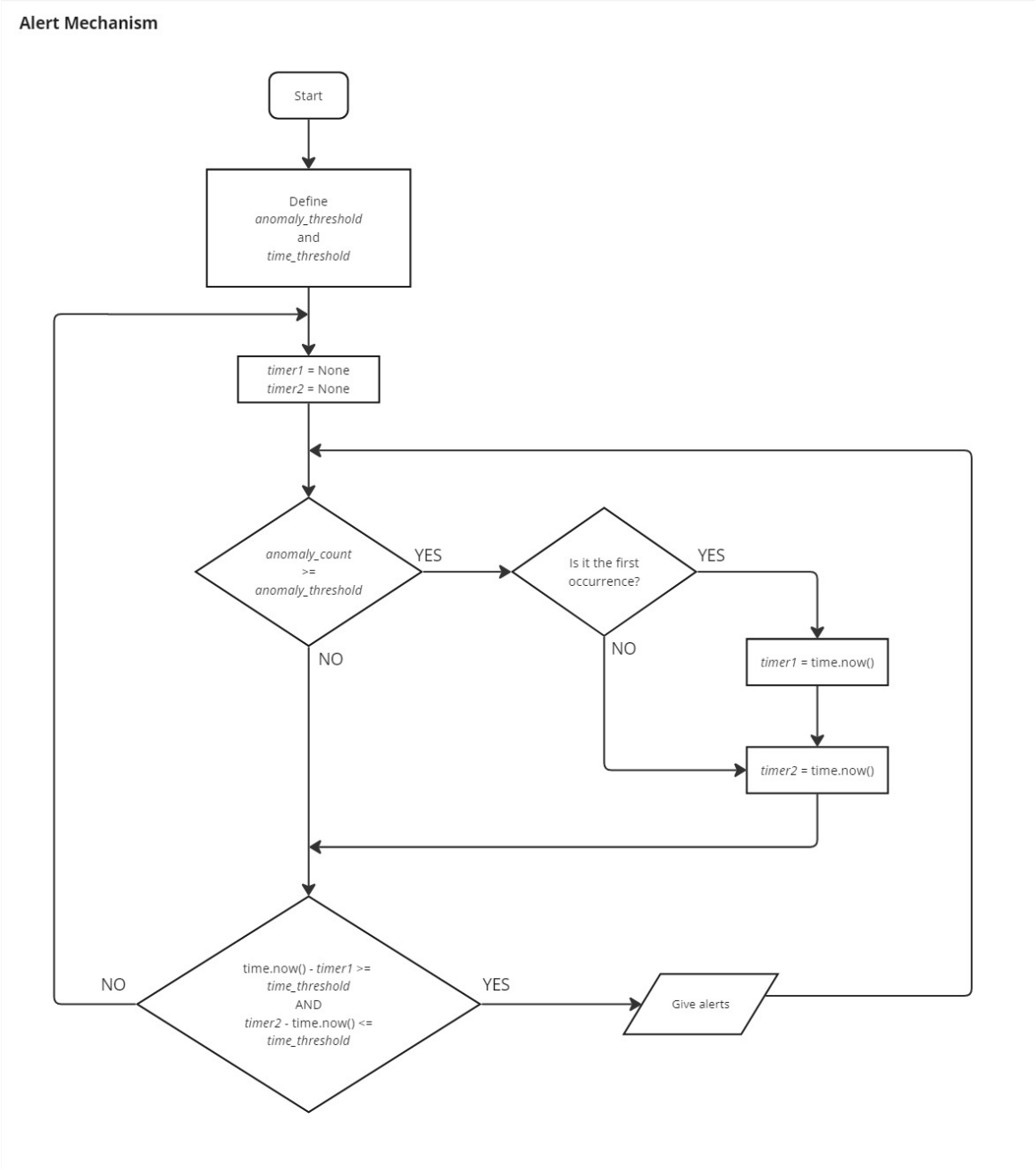


Figure 6.4: Flow diagram of the alert mechanism

Chapter 7

User Application

User Interface is made using the Python tkinter tool [9] which gives a simple robust look to the app. The app is the integration of all the software-level functionalities discussed below.

- USB connection:
  Establishment of the USB connection for serial communication.
- ML:
  With simple clicks, the Application will run the ML model at the backend.
  Training of the ML model can be done with previously connected data sets.
- Visualize:
  Graphs of the real-time data can be visualized using the application.
- Error handling:
  Error handling, show warnings, show information and run-time guidance system is also integrated with the application.
- File System:
  Users can store raw data and trained ML models anywhere on their PC and they can load and store them at the run time of the application. And they can organize these data and models without running the application

The Basic architecture of the app is provided below. It consists of 5 main components File System, Communication, Data Collection, ML model, and Data Visualization.

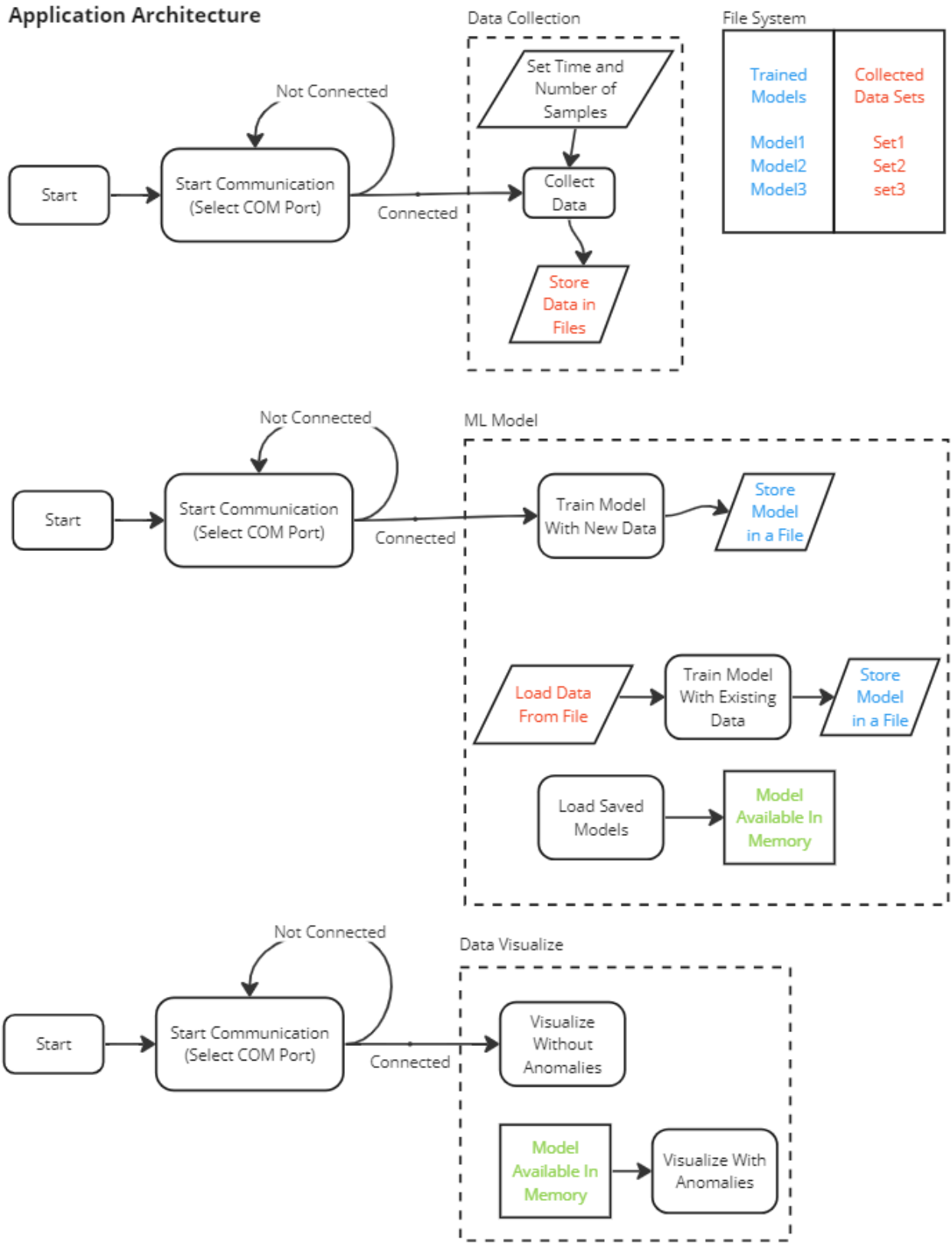


Figure 7.1: Architecture of the application and dataflow

Chapter 8

Data Visualization

8.1 Visualization Overview

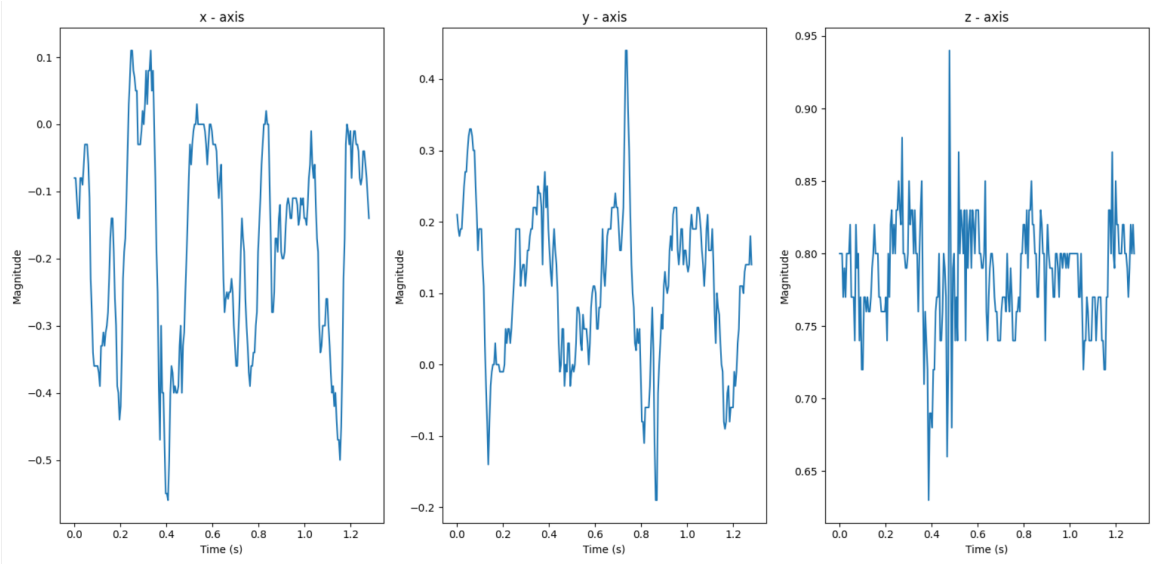


Figure 8.1: Data visualization

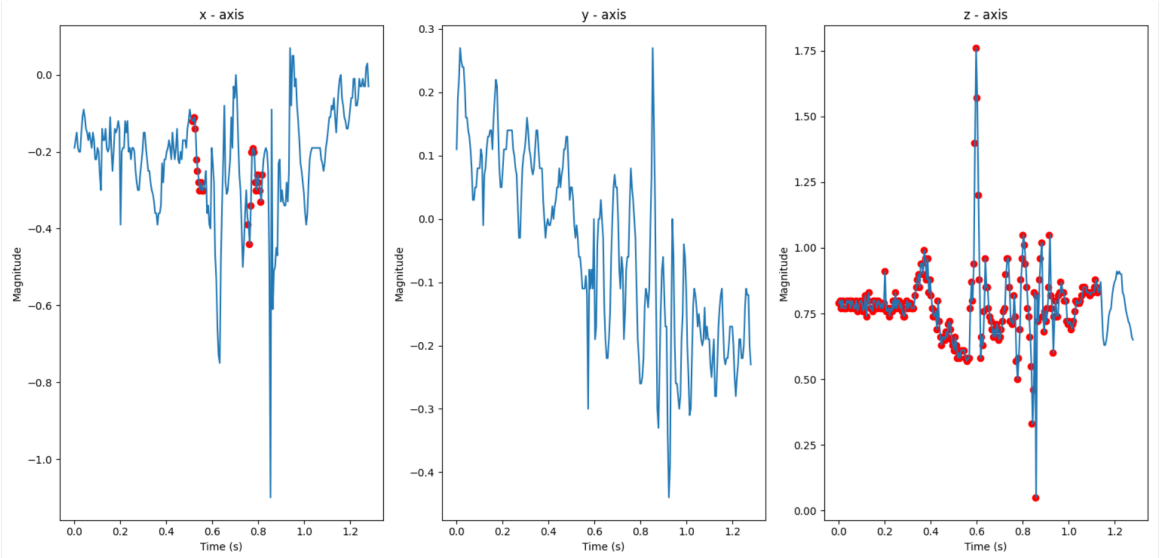


Figure 8.2: Data visualization with anomalies

Visualizing vibrations is an essential part of the project. In that case, vibrations can be visualized as the acceleration data of the three separate axes. There, the user can see and get some insights about the periodic nature and the patterns present in the data.

Real-time continuous data streams can be visualized in two modes.

1. Just the incoming data streams without anomaly detection.
2. Incoming data streams with anomaly detection.

Matplotlib library in Python is utilized for this purpose.[8].

It is integrated with the Application and provides an interactive plot where users can capture pictures of different time steps and critically analyze the anomalies of the vibrations.

The x-axis is time and the Y-axis of the graph is the magnitude of acceleration component of that particular axis.

8.2 Calculations

Data accumulation in the microcontroller is as follows.

Sampling rate (Samples Per Second)

$$= sps$$

$$T_{sample} = \frac{1}{sps}$$

Buffer size

$$= x$$

Time required to fill the buffer (per axis)

$$t_{axis} = \frac{x}{sps}$$

Number of axes

$$= n$$

Total number of samples collected per $t_{axis} time$

$$N = x \times n$$

This $x \times n$ amount of data collected over t_{axis} time needs to be sent to the computer over UART.

UART data rate calculation is as follows. Note that data is stored in the *uint8* format.

Total number of bits collected over time t_{axis}

$$= N \times 8$$

Minimum data rate required

$$r_{min} = \frac{N \times 8}{t_{axis}}$$

The baud rate of the UART communication link should be selected according to this requirement.

Chapter 9

Future Improvements

Our current product architecture and design can be upgraded further by applying the following improvements.

1. Build cloud-based system
2. Build fully modular architecture
3. Improve anomaly detection techniques

To upgrade this product to a cloud-based system, we must address the constraints at different levels such as

- Look for alternative microcontrollers
- Find wireless communication techniques to transfer data to the cloud
- Think about the hosts(for example [10]) and its many other options and try to find their offers and limitations. their support for ML training, inferencing, data processing, DSP and UI.

To Build a fully modular architecture, we must rethink PCB, Communication to data transfer internally from modules to microcontroller, and many more
we found a good industrial approach to this. It is the Connected Area Network(CAN) and related communication protocol. For more information,
you can refer to [11]

There are many types of anomalies related to industrial machine vibration. We can deploy different ML models to detect anomalies to improve anomaly detection techniques. In this case, we have used Autoencoders but there are many other techniques to do it.
In our current approach to detect anomalies we didn't do frequency domain analysis because the autoencoder does not require it. we can use Fourier analysis like FFT to extract more data related to anomalies

Chapter 10

Conclusion

This project has been completed by a two-member group, as a partial, but main requirement for the EN2160 module, in the ENTC curriculum.

In the project, we developed a system to monitor and analyze the vibrations of industrial machines. The system can be identified as a hardware-software co-solution, as it consists of significant aspects of both hardware and software design and development.

Hardware implementation of the project can be identified as having two main parts, namely the main microcontroller unit, and the extendable sensor unit(s). Similarly, software implementation also consists of the user application, machine learning models, and data visualization.

We utilized accelerometers as the main type of sensors to measure vibrations. The microcontroller unit samples the sensor(s) at a constant given sampling frequency and accumulates data in an internal buffer. Once the buffer is full, data is sent to the computer through the UART protocol. Development of firmware is also a part of this process.

The computer is then responsible for handling these incoming data. If only the data visualization mode is turned on, data will be sent directly for visualization. If the anomaly detection mode is turned on, data has to go through trained machine learning models to inference for anomalies.

We also designed PCBs for the microcontroller unit and extendable sensor units. Other than that, enclosures were also designed considering the protection of PCBs as well as easy mounting on a machine. These were done using professionally used software.
Moreover, in the software, the user is given the flexibility to control the system by configuring and training new models for a new machine or a changed setup and to switch between modes mentioned above.

We have implemented an alert mechanism with careful consideration of not giving unwanted and cumbersome alerts while providing essential notifications.

Other than that, the project was all about careful following of the good practices in project management. With all the documentation, data logs, and preliminary studies such as requirement analysis, and stakeholder map, we could adhere to that.

Appendices

Appendix A

Diaries and Design Decisions

A.1 Diary 1 - R.M.L.H. Ratnayake

- **2024-02-14**: Checked the feasibility of doing FFT (Fast Fourier Transform) in the Arduino platform using Arduino Mega.
- **2024-03-14**: Got accelerometer readings through I2C protocol, and performed FFT on them.
- **2024-03-17**: Sampled data considering,
  1. Nyquist's sampling theorem
  2. Use internal timers of ATmega328P to trigger timer interrupts to read accelerometer data.
- **2024-03-27**: Updated code to sample at any given frequency, in the $1\,\text{Hz} - 1\,\text{kHz}$ range. Fine tuned data plotting of the collected data by,
  - Adding axis labels and correct scaling.
  - Using double-sided FFT for more appealing look, and clarity.
- **2024-03-31**: Sent collected data to the computer using UART protocol, plotted graphs of realtime data.
- **2024-04-01**: Worked more on the finer details about visualizing data.
- **2024-04-02**: Adjusted sampling frequency and the number of samples to suit the use case (200 Hz frequency, and 200 samples), Started designing the schematics for the PCBs (two PCBs for the MCU, and the accelerometer sensor).
- **2024-04-03**: Started using PyCharm IDE for easy development, continued on PCB schematics.
- **2024-04-04**: Finalized PCB schematics with almost no mistakes, and double checked.
- **2024-04-05**: Finalized PCB schematics, start placing components in the PCB layout.
- **2024-04-08**: Started routing between components in the PCB layout designs.
- **2024-04-10**: Coded the program to receive data continuously that are being sent by the MCU.
- **2024-04-12**: Tried to use "probability" method to detect anomalies, created some basic functions needed to perform it.
- **2024-04-13**: Finalized placing and routing of the PCBs of both MCU and sensor.
- **2024-04-16**: Completed the redundant PCB design of CH340G, decided to use "autoencoder" method and did some minor changes in data acquisition and pre-processing to suit the need.
- **2024-04-17**: Generated gerber files and placed the order in JLCPCB.

- **2024-04-19**: Created a simple ML model to detect anomalies in x-axis only.
- **2024-04-20**: Tried to extend the same model to all three axes and failed with an error with dimension mismatch.
- **2024-04-21**: Deliberately distorted some of the collected data by adding random values, in order to test whether the distortion is detected by the model as an anomaly.
- **2024-04-22**: Restructured the code to use three models for the three axes, placed an order in Mouser Electronics for electronic components, with some other teams.
- **2024-04-23**: Wrote a code to combine the data collection and ML parts (training and inferencing) together, thus achieving a milestone in the project. After this developing the software architecture becomes somewhat straightforward.
- **2024-04-24**: Trained the ML models and it was successful. Decided to use multiprocessing to enhance performance and reduce time to train.
- **2024-04-26**: Updated plotting to only plot time domain data.
- **2024-04-27**: Plotted with real-time detected anomalies on the existing graphs, PCBs arrived and checked for the availability of the ordered products: two PCBs, and a stencil.
- **2024-04-30**: Soldered and tested both PCBs with the help of the other teammate.
- **2024-05-01**: Saved collected data in the *uint*8 format in the microcontroller to save up space.
- **2024-05-02**: Coded different modes of training the ML models (e.g. from new data, from existing data, load saved models from file).
- **2024-05-03**: Coded the alert mechanism for detected anomalies.
- **2024-05-04**: Improved GUI look and feel, functionality, and user-friendliness.
- **2024-05-05**: Got some collaborative decisions regarding the final design for the enclousure, with the other teammate.
- **After 2024-05-10**: Preparation of design documents and reports, careful re-evaluation and double-checking or all the aspects of the project.

A.2 Diary 2 - N.P.S.S. Rupasinghe

- **Before 2024-03-01**: Understanding the design methodology, and user requirements, and analyzing the market about the ideas related to our design.
- **From 2024-03-01 to 2024-03-10**: Got a Microsoft Azure VM, adjusted its settings, and tried to implement a fully online mode Database with a processing part. Found many limitations there.
- **2024-03-12**: Finalized the documentation part of phase 2.
- **2024-03-15**: Finalized the Conceptual design (Fully wired communication for functional block diagram (USB, I2C) part and hybrid Modular design architecture).
- **2024-04-02**: Finalized the PCB Schematics, and the design SolidWorks part, and created drawings about the dimensions of the design.
- **2024-04-03**: Finalized the PCB Schematics – PCB was finalized with FTDI USB to UART converter.
- **2024-04-05**: Found some errors in FTDI and decided to change the PCB with CH340 Module components (already used in Arduino). Discussed it with other groups. Decided to use TKINTER GUI with Python backend for the application. Tried to find a motor that vibrates at the department - UAV lab grinding motor – discussed it with the lab assistant in the UAV lab. Worked on finding more solutions.
- **2024-04-08**: Finalized the appearance of the product sizes for the SolidWorks design, chose to go with the suction cup mounting method, chose the wires to be used in the device to connect modules.
- **2024-04-11**: Found a tkinter-based app to create GUI MATDECK in LABDECK (https://labdeck.com/python-designer/tkinter-gui-designer/). Found that it does not work (has to purchase a paid version) and moved on with basic tkinter, matplotlib, and python serial libraries. Got help from the online tool (https://visualtk.com/).
- **2024-04-15**: All the necessary functionalities are added to the app including deleting folders, creating folders, checking real-time workings, and testing. Made the architecture of the app a normal app with no OOP concepts (hard to do real-time tasks with OOP).
- **2024-04-19**: Found a motor with a slightly misaligned axle (ideal for our prototype testing verification and demonstration). Made orders from the local market for ports and some elements, hoping to get the PCB by 2024-04-26.
- **2024-04-21**: Prepared the item for demonstration. Discussed with the teammate how to be unique – ML, and Modular architecture. Considered many anomaly-detecting techniques like 10 different ML models and threshold methods.
- **2024-04-25**: Integrated the Application UI with other codes - ML, Data sampling, Com port detection, and Data plotting.
- **2024-04-27**: Made a website for the product to include all documents and host the application and updates (http://vibroguard.unaux.com/).
- **2024-04-28 to 2024-05-03**: Tried different user interfaces and integrated all functionalities with the UI.
- **2024-05-04**: Finalized the enclosure sizes and all parts fully defined sketches with all the relevant components except the mounting boss to fit the lid and base. Tried to find a compatible port - ETHERNET port or port that we bought.

- **2024-05-08**: Sent the required files to print the enclosure.
- **2024-05-11**: Finalized the application architecture and UI with buttons and organized the documentation such as how to write preliminary design and comprehensive design documentation.
- **2024-05-15**: Worked on a few bugs in the code. The ML part was not responding due to a missing functional block on the interface.
- **From 2024-05-11 to 2024-06-12** (Non-Academic Strike):
  Dedicated this period to improving design documentation.
  Added appendices to enhance readability.
  Finalized the preliminary design document
  Finalized comprehensive design document.

A.3 Design Decisions

2024-02-07 - Use frequency domain data to detect anomalies.
Reason:

It would be easier to perform FFT on incoming data and use a common ML algorithm to detect the changes in the magnitudes of the frequencies present in the spectrum.

2024-03-25 - Plot data to get visual feedback.
Reason:

When plotting the acceleration data, it is easy to get insights whether the project is progressing. Such insights include:

- Accuracy of the data from the sensors.
- Whether the sensors are working as expected.
- The user can also be given a good idea about the vibrations.

2024-03-31 - Decided to create a GUI application using Python.
Reason:

To combine all the aspects and parts of the project such as data acquisition, visualization, training ML model, predicting/ detecting anomalies etc. and to give the user a all-in-one control interface. Python is specially selected since the GUI is not a very complex or fancy-looking interface as well as it is easier to interface with existing functions specially regarding ML.

2024-04-01 - Use FT232RL as USB-to-UART converter.
Reason:

FT232RL USB-to-UART IC is selected to be used in the PCB to send acceleration data to the computer. This decision is taken considering the availability in the market, and the good reputation of the FTDI chips.

2024-04-06 - Design PCB with CH340G in case that the PCB with FTDI does not work.
Reason:

In common market-available products where ATmega328P is used as the MCU, CH340G USB-to-UART converter chip is used. But the chip is not available in both local and international market to be bought. So, readily available FT232RL chip by FTDI is chosen for the main PCB. However, alternative PCB with CH340G is planned to be designed in case FT232RL did not work.

2024-04-10 - Use "probability" method to detect anomalies.
Reason:

This is a very simple model in ML and can be implemented easily with our current knowledge.

2024-04-18 - Use autoencoders to detect anomalies.
Reason:

The "probability" method has several drawbacks.

1. Requires a dataset that is already classified as anomalous and normal, which is difficult to achieve in a realtime environment.
2. The method is best used with frequency domain data. But this involved additional computational power (FFT) and can be constrained with the capabilities of the computer.

Hence, autoencoders are used and they have the following advantages.

1. Can use time series data directly without converting to frequency domain.
2. Use din the industry more often to detect anomalies in various situations.
3. Easier to find resources on how to use autoencoders.

2024-04-22 - Use three copies of the same ML model for the three axes.
Reason:

Attempt to scale to autoencoder model to take the data of all three axes together is failed. As a result, it is decided to use three copies of the same model separately for the three axes.

2024-04-25 - Use multiprocessing.
Reason:

Use Python's inbuilt feature of multiprocessing to train the ML models of the three axes. It uses all available cores of the CPU, and hence increase the speed of training.

2024-04-26 - Multiprocessing is not used for inferencing.
Reason:

Multiprocessing will not be used in predicting anomalies (inferencing).

- It is not as computationally intensive as training.
- Takes much time than the normal (single processor core) method, probably because it requires copying of a lot of data (data collected from sensors) between processes.

2024-04-27 - Plot FFT graph as an addition.

Reason:

If time permits and computer resources are sufficient, frequency domain graphs of the incoming data can also be plotted.

2024-04-30 - No PCB with CH340G, because the PCB with FT232RL works fine.

Reason:

PCB with FT232RL is soldered and tested, and it started working successfully as expected. Hence, no need for a PCB with CH340 to be used.

2024-05-01 - Use $uint8$ data type to store collected data in the MCU.

Reason:

Previously, accelerometer data were stored in the buffer as they receive as $float32$ which takes 4 bytes each. Since ATmega3228P has only 2KB of memory, it is decided to use $uint8$ (unsigned integer - 8 bits). The data are first multiplied by 100, and mapped to 0 - 255 range before storing. Then, once they are sent through UART to the computer they are again decoded back to their original type.

However, this involves some accuracy loss, but it is tolerable with the design specifications.

Appendix B

User Guidance

This document serves as a comprehensive guide to help you set up, operate, and maximize the functionality of the Vibration Analyser. The Vibration analyzer is a sophisticated tool designed to analyze and monitor vibrations in mechanical systems. This manual covers the setup and usage of the hardware (main controller and modules) and the accompanying desktop application.

B.1 Hardware Setup

B.1.1 Enclosure

Depending on your product (Your vibrating device) we will give you a chance to make your enclosure because this enclosure is designed for flat surfaces with screw mounting enabled. For some vibrating devices the machine is designed in different shapes and conditions. Users have to make educated decisions on how to connect the vibration monitoring system to their devices. Your design has to be compatible with our PCB and our modular architecture.

Guidance or methods to create your design

Positions to connect

- Main Controller - outside the vibration device near your personal computer
- Module - attach to your vibrating device

How to connect

- Suction cups - use standard suction cups
- magnets - you have to consider how it affects the PCB
- Screws - preferred

Things to Consider

- No external power supply is needed
- Powered by computer via USB
- Power is indicated by the LEDs on the Main controller and Modules

B.1.2 Main Controller Connection

You can connect the main controller to your computer using the USB cable. (USB B to USB A port)

Figure B.1: USB connection for main controller and computer

B.1.3 Module Connection

You can connect up to 2 modules on top of each other (make sure to connect them where you assume faults can occur) Ensure that the modules are securely connected to your vibrating device to prevent any loose connections. Securely wired the modules to prevent loose connection because it may affect reliable data communication

Connect two modules using the provided mechanical ports (for strong connection).

wiring convention and other details are provided under the enclosure design chapter.

Figure B.2: Industial M12 4 pin connector connect moduels

B.2 Software Setup

B.2.1 System Requirements

Verify that your computer meets the minimum system
requirements for the desktop application.

Processor:
Intel Core i3 (8th generation or newer)

At least quad-core for optimal performance
Clock speed of 2.5 GHz or higher

Memory (RAM):
Minimum: 8 GB RAM for basic usage
Recommended: 16 GB RAM for multitasking and running resource-intensive applications

Storage:
Solid State Drive (SSD) is preferred for faster boot-up and application loading times
Minimum: 1 GB SSD for storage
Recommended: 5 GB SSD or higher for ample storage space on SSD

Windows 10 (64-bit) is recommended for compatibility with the latest software and drivers

B.2.2 Installing Steps

- Take the executable file from the SD card provided with the device or download it from the website http://vibroguard.unaux.com/
- Double-click the executable file to initiate the installation process.
- Follow the on-screen instructions to complete the installation.

For more details and new apps made for the same device and to updates for apps visit our website

B.3 How to use the Application

Here is an entire explanation for using the application

B.3.1 Launching the Application

Locate the installed application icon on your desktop or in the Start menu. The icon is given here. Familiarize yourself with the different sections and features of the desktop application.

Figure B.3: Application Icon

Double-click the icon to launch the application.

B.3.2 UI guide

Start Communication

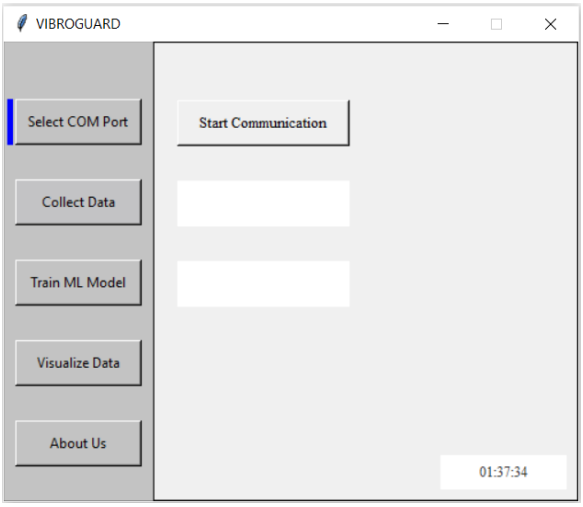


Figure B.4: image-1

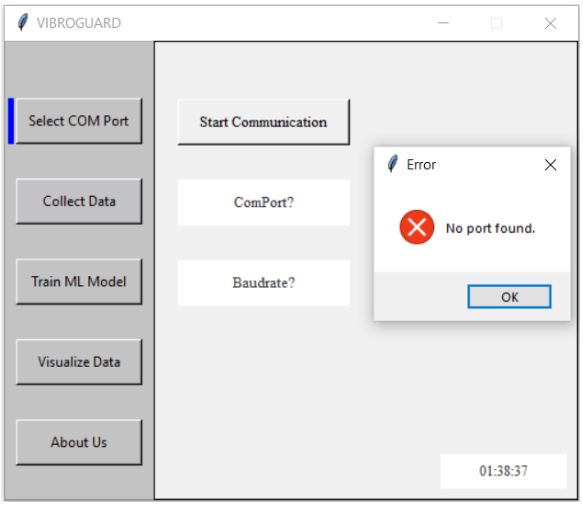


Figure B.5: image-2

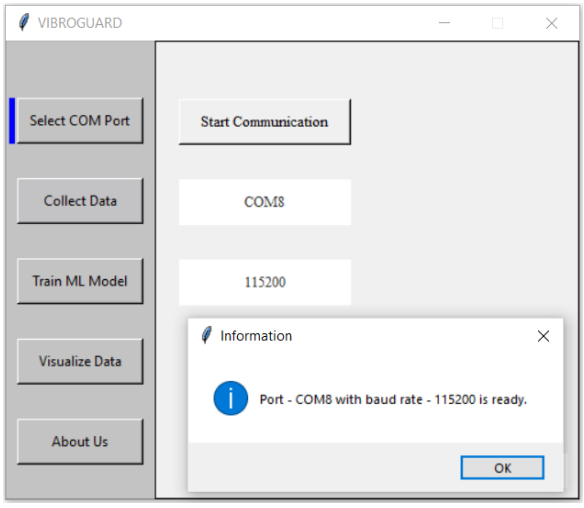


Figure B.6: image-3

- image-1 = Beginning
- image-2 = Try to start communication without connecting the main controller to the PC
- image-3 = Connection without errors

Collect Data
You can collect data after starting the communication.

without connecting modules to the main controller, the stand-alone main controller will generate and collect junk data.
make sure to connect at least one module before collecting data.

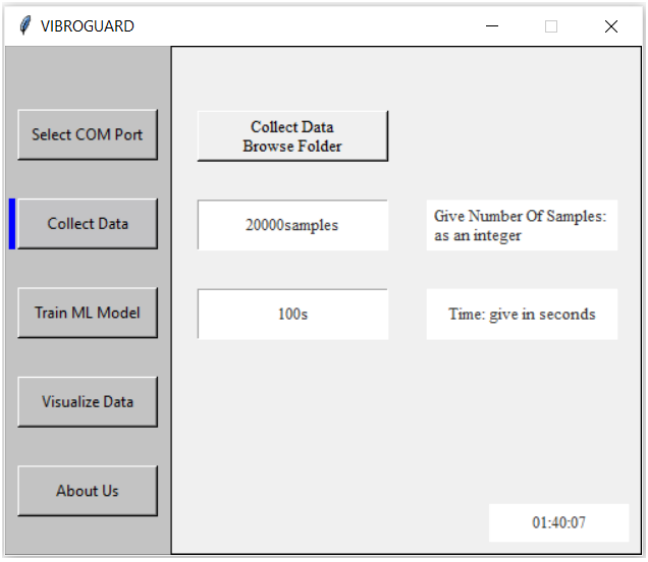


Figure B.7: image-4

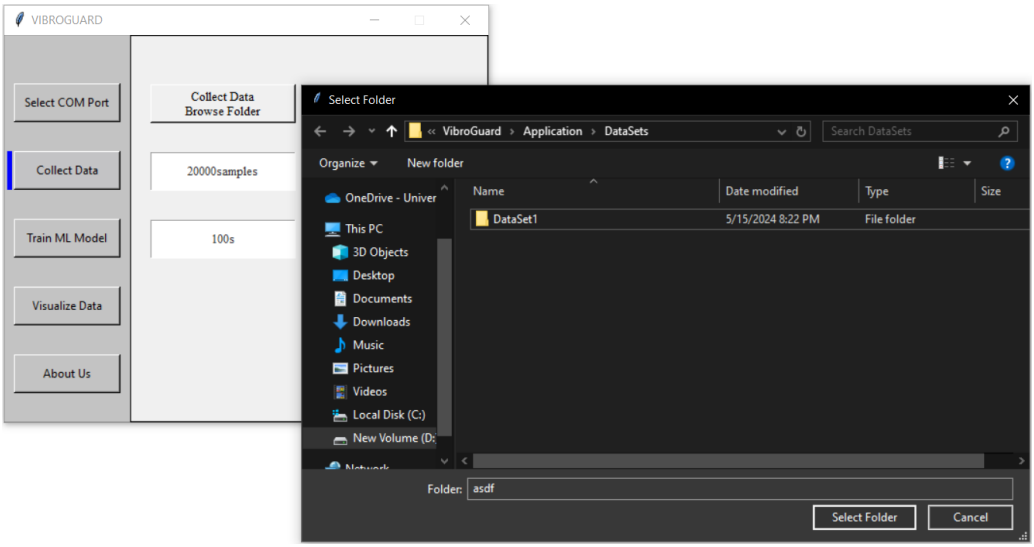


Figure B.8: image-5

- That page contains 2 thresholds. Users can change it.
- When click the button you can select where you want to store collected data.

ML model

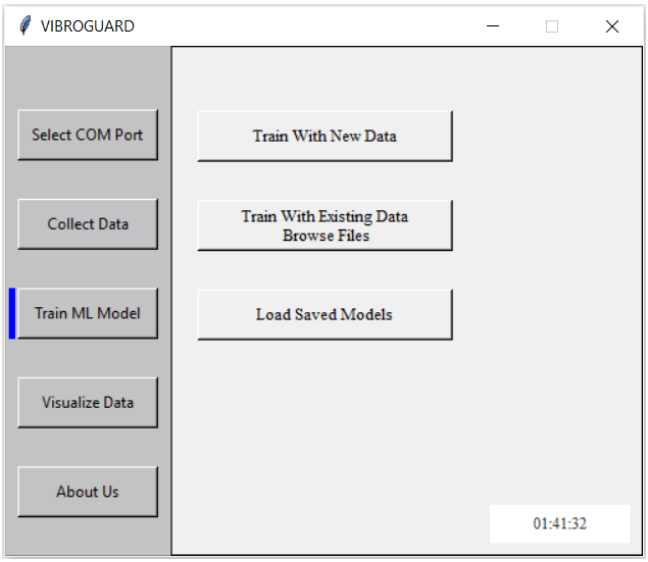


Figure B.9: image-6

- Users can train ML models from newly collected data, Make sure the connection is first to do so. Then They can save the model anywhere they want.
- Users can train ML models from previously collected data. Make sure that collected data exists on the PC. Then They can save the model anywhere they want.
- Before using the ML model, users have to Load it from where they have saved it.

Visualize

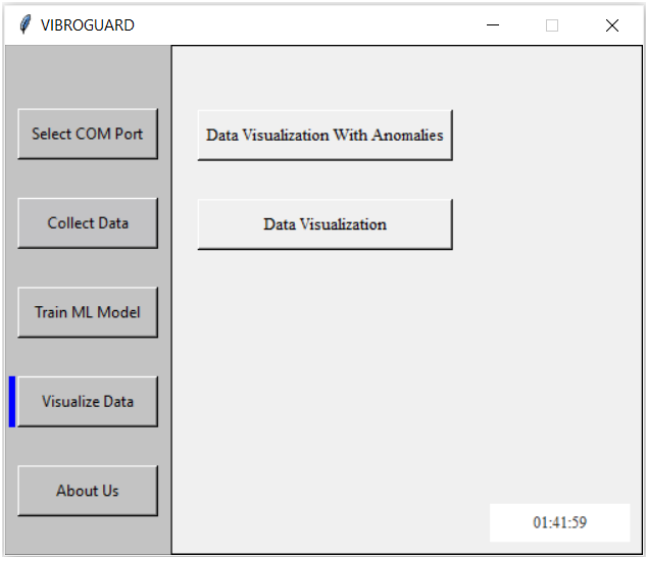


Figure B.10: image-7

- Users can visualize real-time data with or without anomalies
- Any way first establish the communication.
- If they want to visualize anomalies Load the model first (image-6)

Website

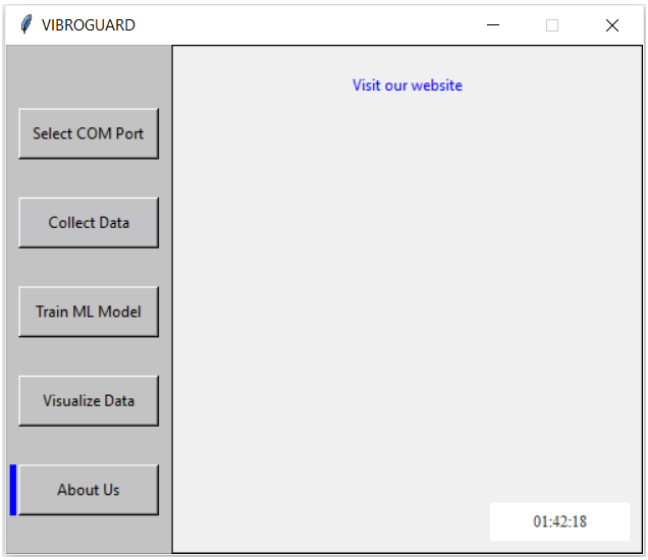


Figure B.11: image-8

- A link to the website is provided on this page.

B.4 Dissect the product

Whether it is the main controller part or the module part, They both have the same design

1. Remove 4 tips that cover the holes of the screws.
2. Use a 3mm screwdriver to remove 4, 3mm screws.
3. Pull the lid from the base.

   Note: Do not pull hard. Be careful with the LEDs and JST connectors attached to the PCB.
4. Remove JST connections.
5. Loosen the M12 4-pin connector and remove it from the enclosure.

   Note: In both enclosures, PCBs are connected with 3mm screws.
6. Remove PCBs by unscrewing the 3mm screws that attach them to the enclosure using a screwdriver.

   Note: Be careful with the PCB of the main controller because the USB port extends through a hole in the enclosure.

Appendix C

PCB Photographs

This appendix contains pictures of the PCB received from JLC PCB and soldered PCBs are also included here.
The PCBs are manufactured and ordered from JLC PCB[15]

Main Controller PCB

- Unsoldered

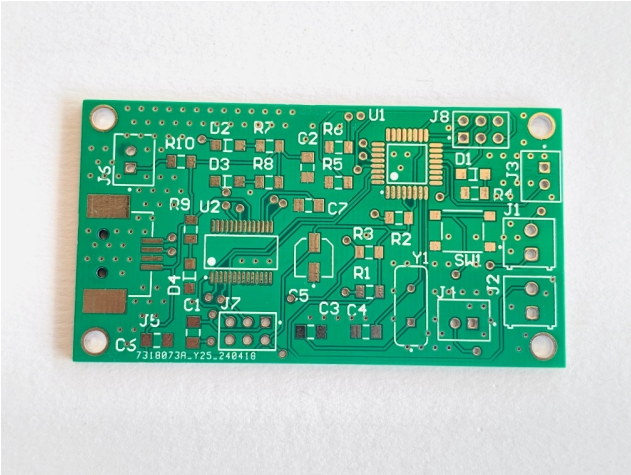


Figure C.1: Main Controller Front unsoldered

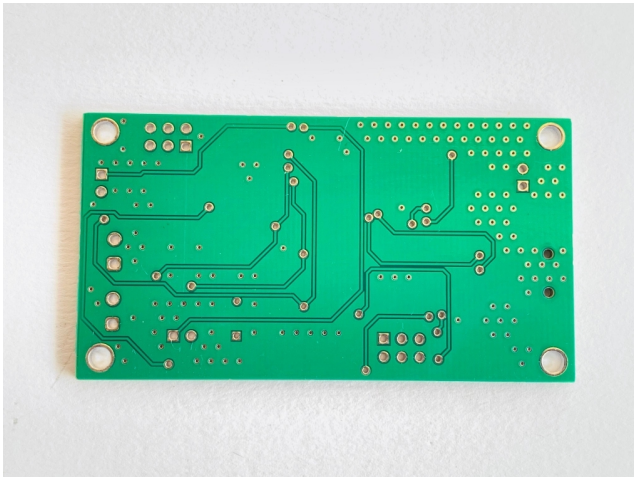


Figure C.2: Main Controller Back unsoldered

- Soldered

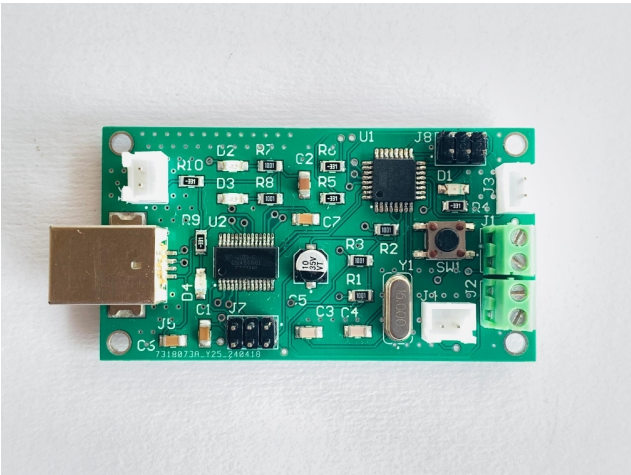


Figure C.3: Main Controller Front soldered

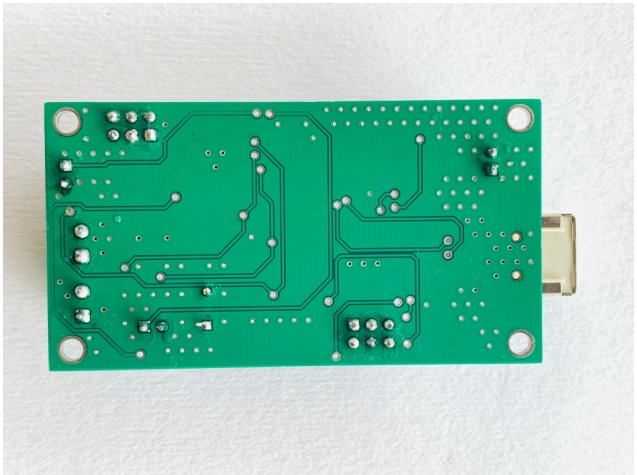


Figure C.4: Main Controller Back soldered

Sensor Module PCB

- Unsoldered

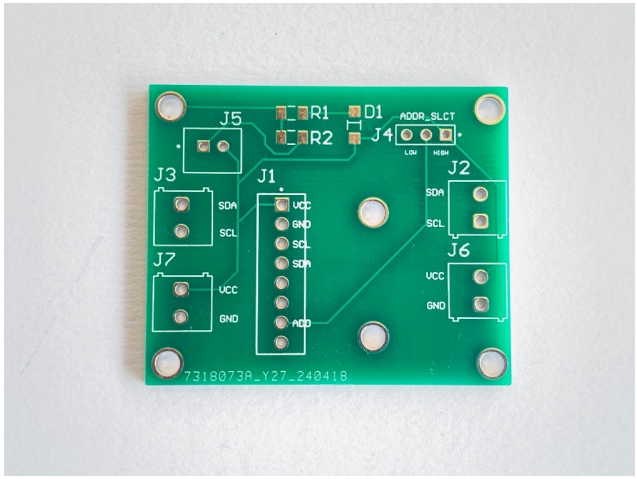


Figure C.5: Module Front unsoldered

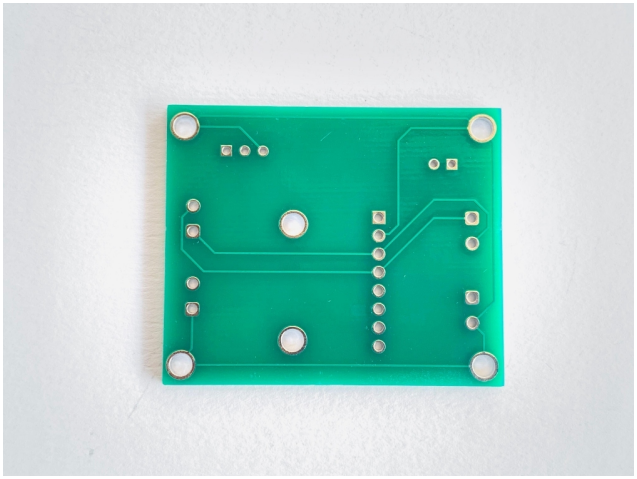


Figure C.6: Module Back unsoldered

- Soldered

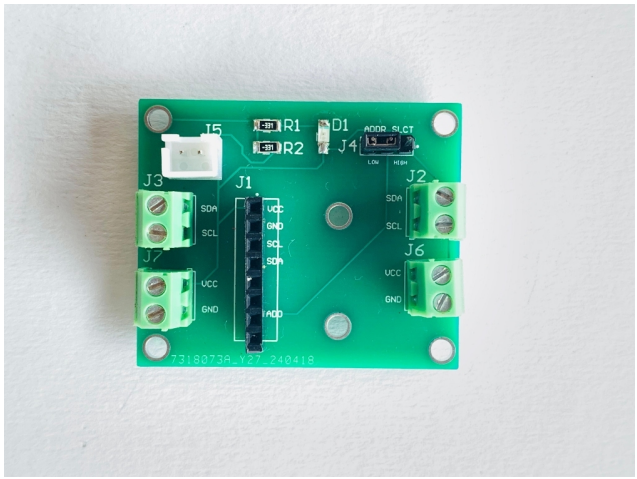


Figure C.7: Module Front soldered

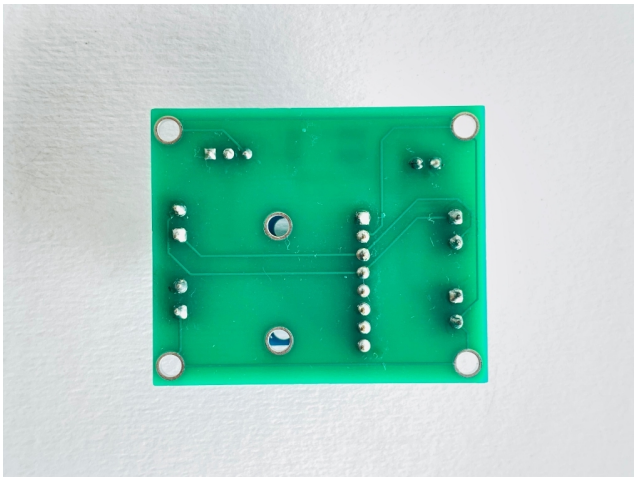


Figure C.8: Module Back soldered

MPU connected Module PCB

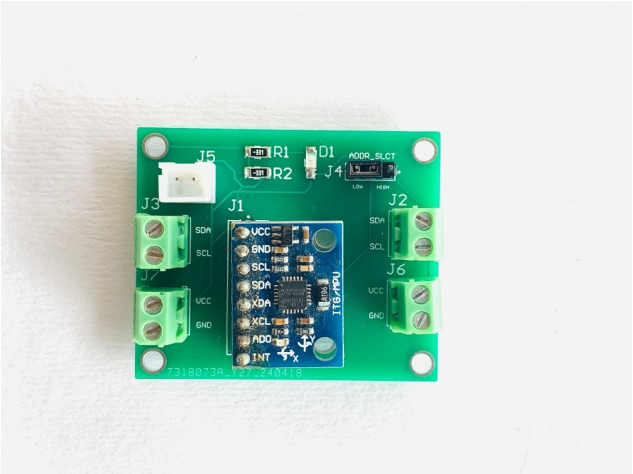


Figure C.9: Module with MPU

Working PCBs

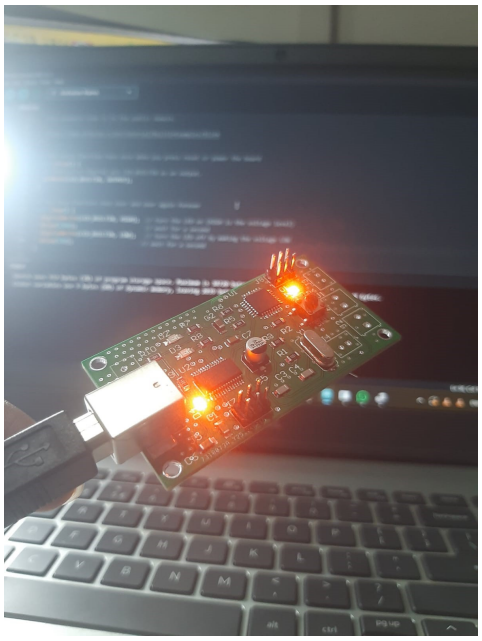


Figure C.10: Working Main Controller

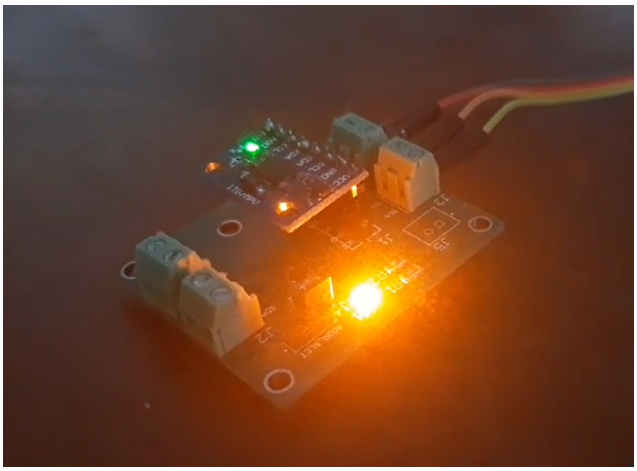


Figure C.11: Working MPU

Unsoldered PCBs Full View - Base

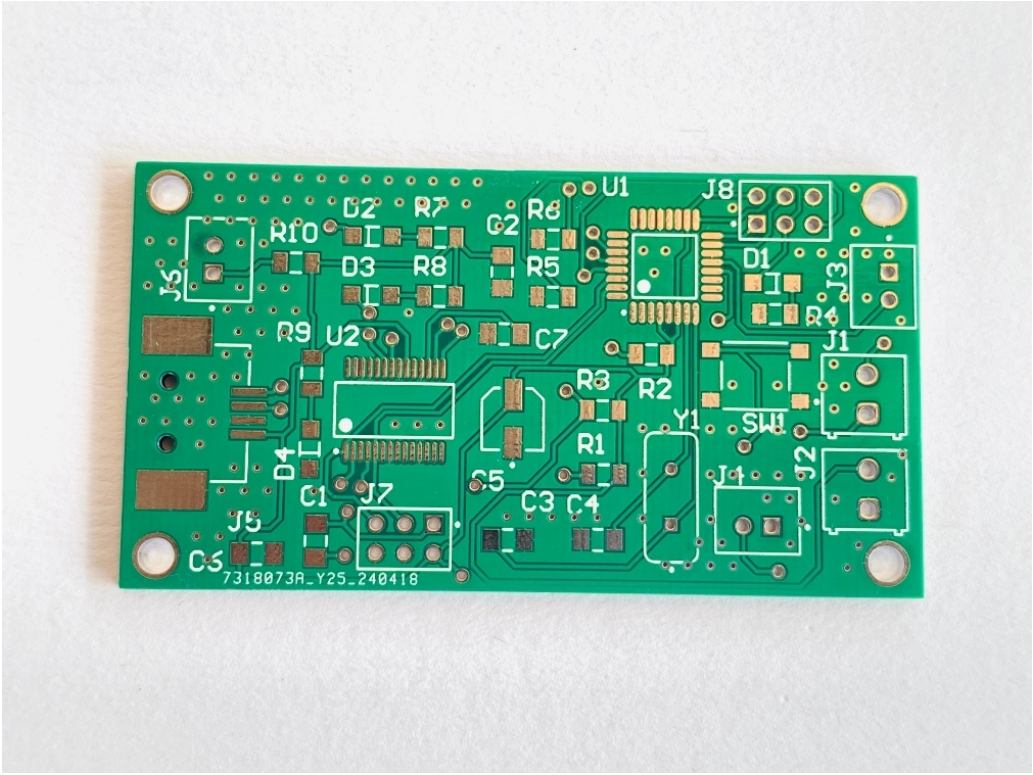


Figure C.12: Unsoldered PCBs Full View - Base

Soldered PCBs Full View - Base

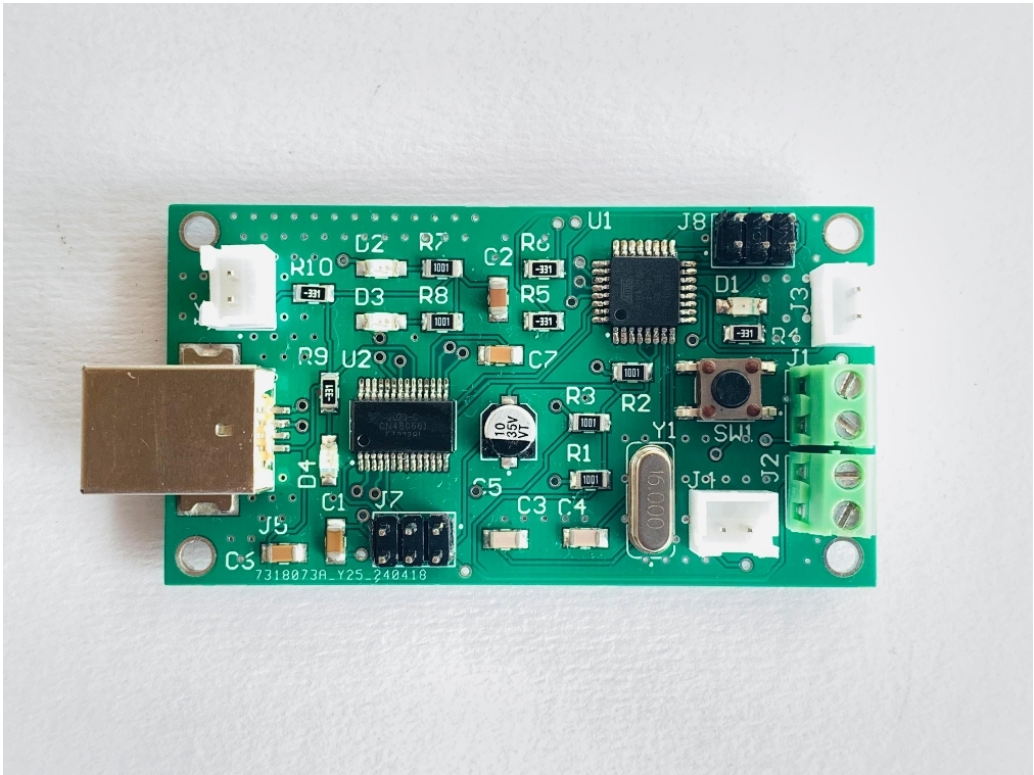


Figure C.13: Soldered PCBs Full View - Base

Unsoldered PCBs Full View - Module

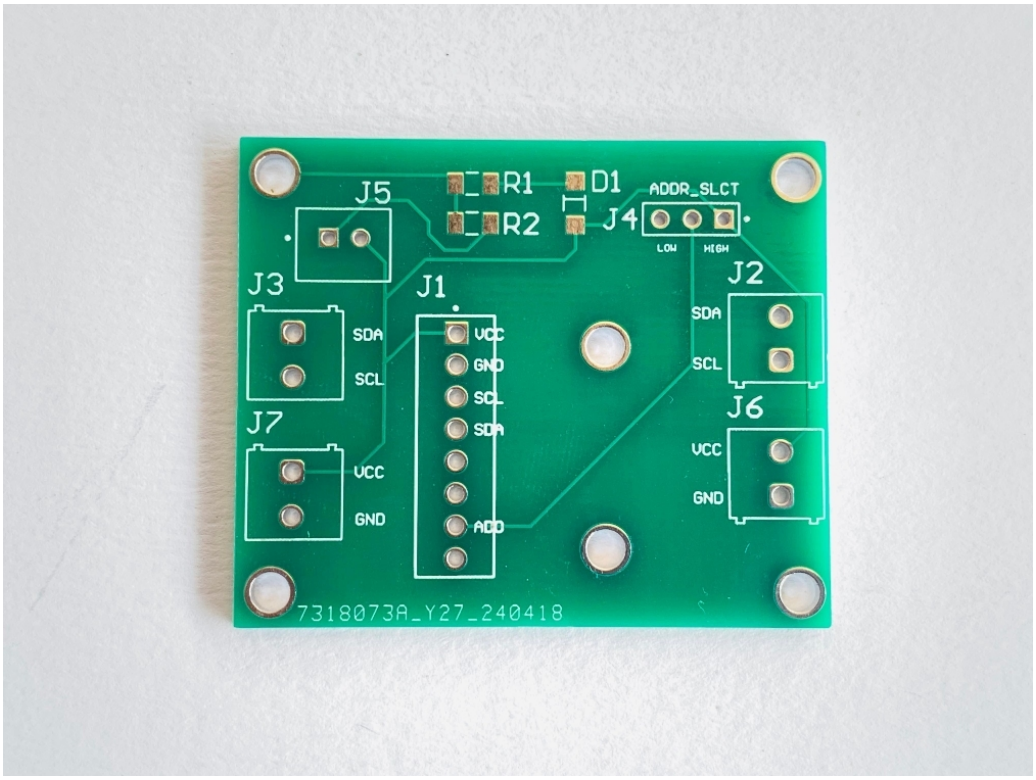


Figure C.14: Unsoldered PCBs Full View - Module

Soldered PCBs Full View - Module

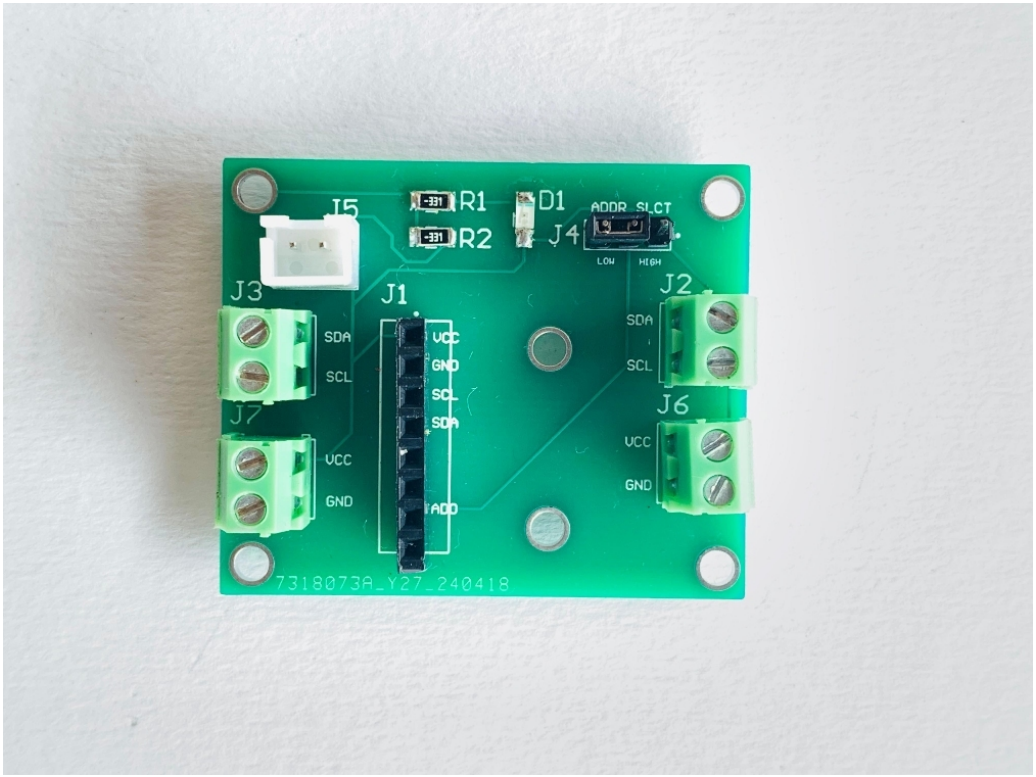


Figure C.15: Soldered PCBs Full View - Module

PCB Sizes(Pictorial manner)

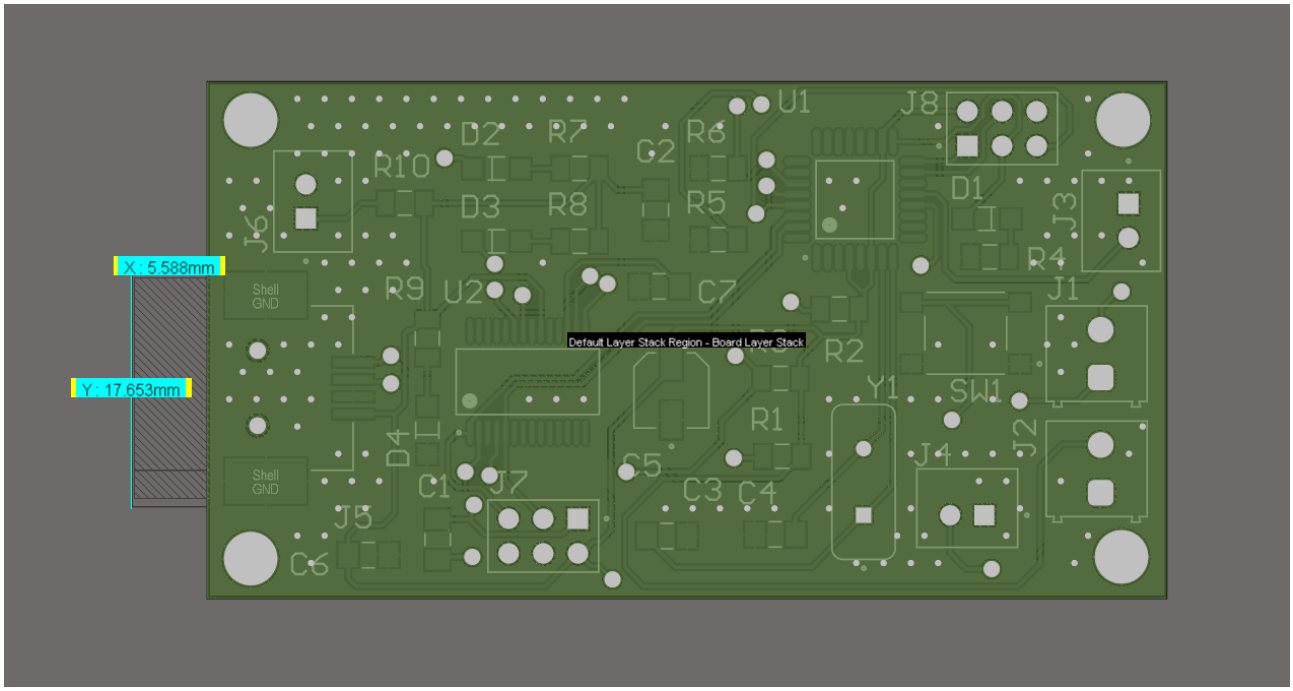


Figure C.16: Sensor PCB (3D view)

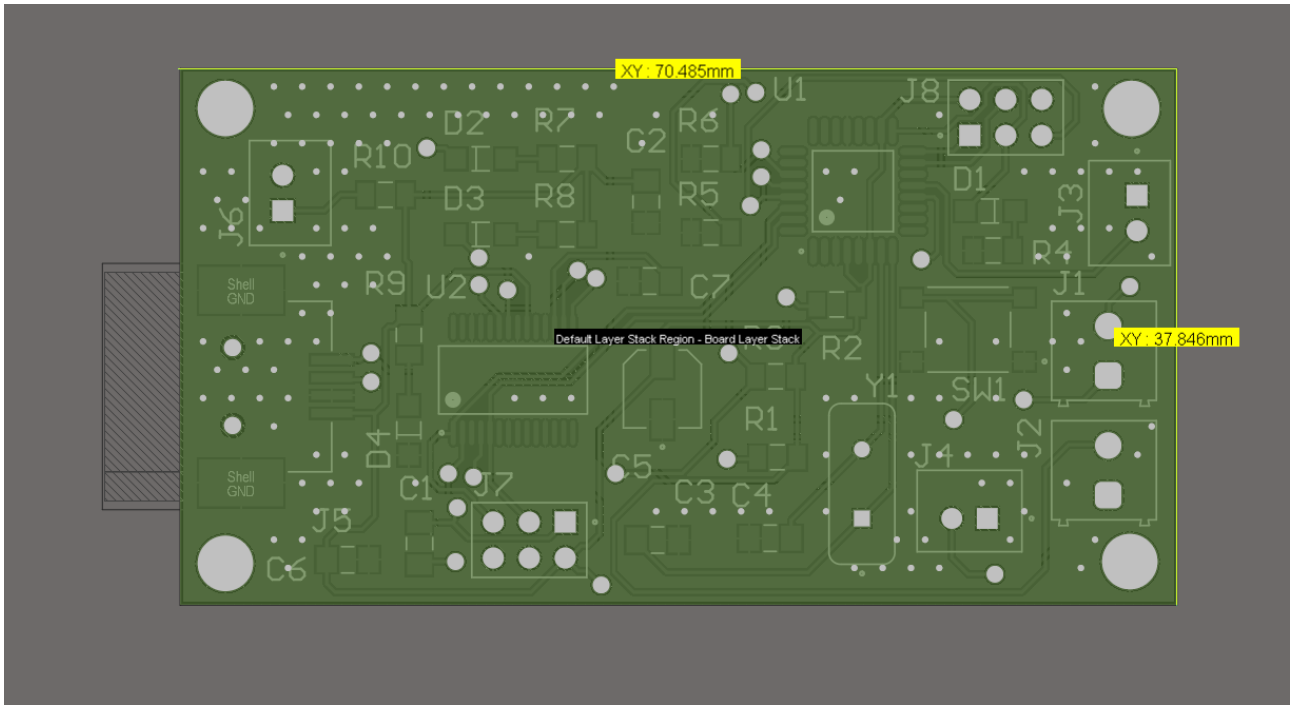


Figure C.17: Sensor PCB (3D view)

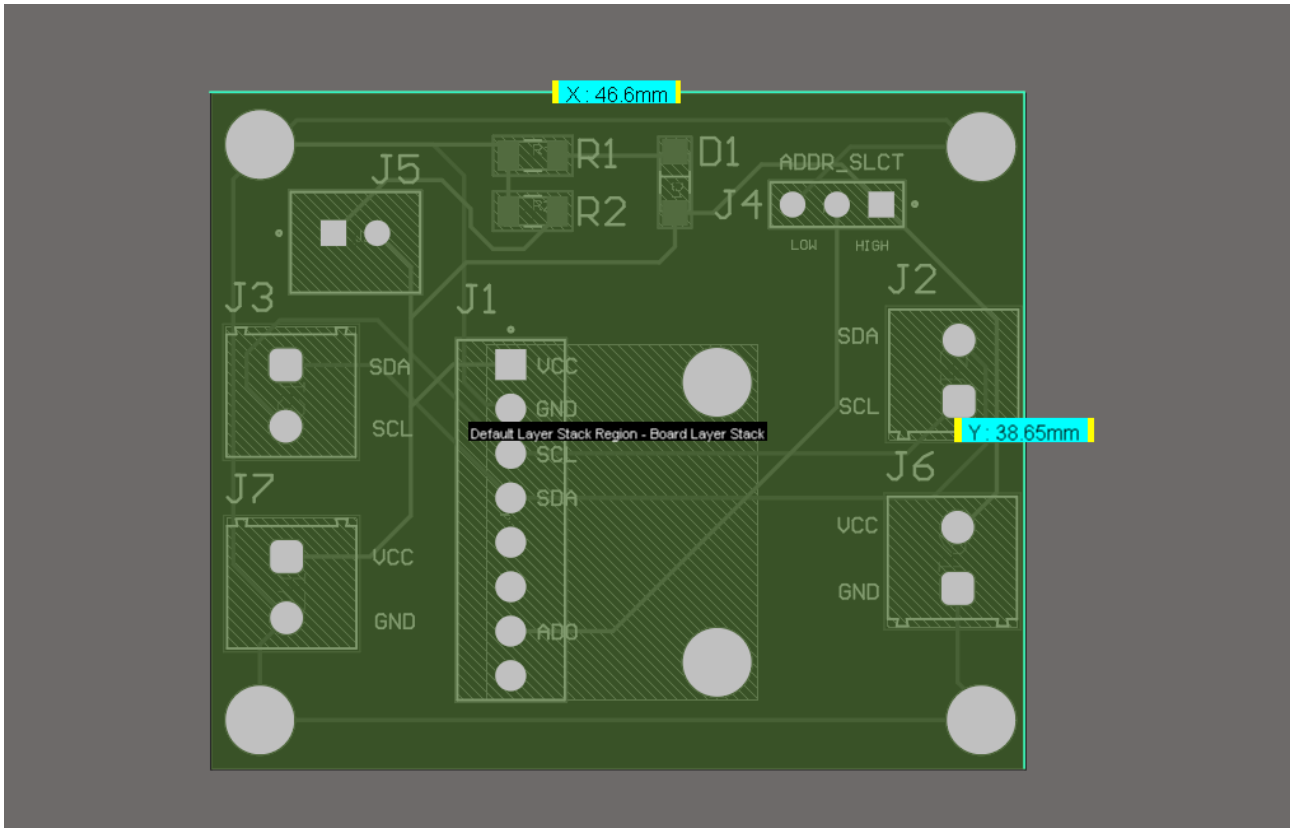


Figure C.18: Sensor PCB (3D view)

Appendix D

BOM

The BOM documents for the 2 PCBs that we have built are given below.
They are organized as follows.

1. BOM of Main Controller
2. BOM of MU (MPU6050 sensor) mount

most of the components are ordered from Mouser Electronics [14]

Comment	Description	Designator	Footprint	LibRef	Quantity
GRM31M1X1H104JA01L	Chip Multilayer Ceramic Capacitors for General Purpose, 1206, 0.10uF, SL, +350/-1000ppm/ °C, 5%, 50V	C1, C2, C3, C4, C6, C7	FP-GRM31M-0_15-IPC_A	CMP-06035-007547-1	6
EEE1VA100SR	CAP ALUM 10UF 20% 35V SMD	C5	FP-SC-MFG	CMP-05427-000087-1	1
HSMG-C150	LED GREEN DIFFUSED 1206 SMD	D1, D2, D3, D4	FP-HSMG-C150-MFG	CMP-2000-05061-2	4
691214110002		J1, J2	691214110002	CMP-1502-02464-2	2
B2B-XH-A(LF)(SN)	CONN HEADER VERT 2POS 2.5MM	J3, J4, J6	FP-B2B-XH-A_LF_SN-MFG	CMP-2000-05888-3	3
62910416121		J5	62910416121	CMP-1502-03244-2	1
M20-9980346	CONN HEADER VERT 6POS 2.54MM	J7, J8	FP-M20-9980346-MFG	CMP-2000-06636-2	2
MCR18EZPF30R0	Thick Film Chip Resistor, 1206, 30Ω, 1%, 100ppm/°C, 0.25W, 200V	R1, R2, R3, R4, R5, R6, R7, R8, R9, R10	FP-MCR18-IPC_B	CMP-08839-012195-1	10
PTS645SM43SMTR92LFS	SWITCH TACTILE SPST-NO 0.05A 12V	SW1	FP-PTS645SM43SMTR92LFS-MFG	CMP-2000-05160-2	1
ATmega328P-AU	8-bit AVR Microcontroller, 32KB Flash, 1KB EEPROM, 2KB SRAM, 32-pin TQFP, Industrial Grade (-40°C to 85°C)	U1	32A_M	CMP-0095-00269-2	1
FT232RL	IC USB FS SERIAL UART 28-SSOP	U2	FTDI-SSOP-28_M	CMP-2000-05029-1	1
FOXSLF/160-20	Resistance Weld Thru-Hole Crystal, 16 MHz, -20 to 70 degC, 2-Pin THD, RoHS, Bulk	Y1	FOX-FOXSLF160-20	CMP-1749-00003-1	1

Comment	Description	Designator	Footprint	LibRef	Quantity
HSMG-C150	LED GREEN DIFFUSED 1206 SMD	D1	FP-HSMG-C150-MFG	CMP-2000-05061-2	1
22-29-2081	22-29-2081 KK® 254 Wire-to-Board Header, Vertical, with Friction Lock, 8 Circuits, Gold (Au) Plating	J1	FP-22-29-2081-MFG	CMP-00257-00277737-1	1
691214110002		J2, J3, J6, J7	691214110002	CMP-1502-02464-2	4
90120-0763	CONN HEADER VERT 3POS 2.54MM	J4	FP-90120-0763-MFG	CMP-04776-000084-1	1
B2B-XH-A(LF)(SN)	CONN HEADER VERT 2POS 2.5MM	J5	FP-B2B-XH-A_LF_SN-MFG	CMP-2000-05888-3	1
MCR18EZPF30R0	Thick Film Chip Resistor, 1206, 30Ω, 1%, 100ppm/°C, 0.25W, 200V	R1, R2	FP-MCR18-IPC_B	CMP-08839-012195-1	2

Appendix E

Enclosure Design and Photographs

This Appendix contains

1. SW Hierarchy
2. Draft Analysis
3. Pictures Of printed enclosure (With and without Components)

E.1 SolidWorks Hierarchy

Main Controller

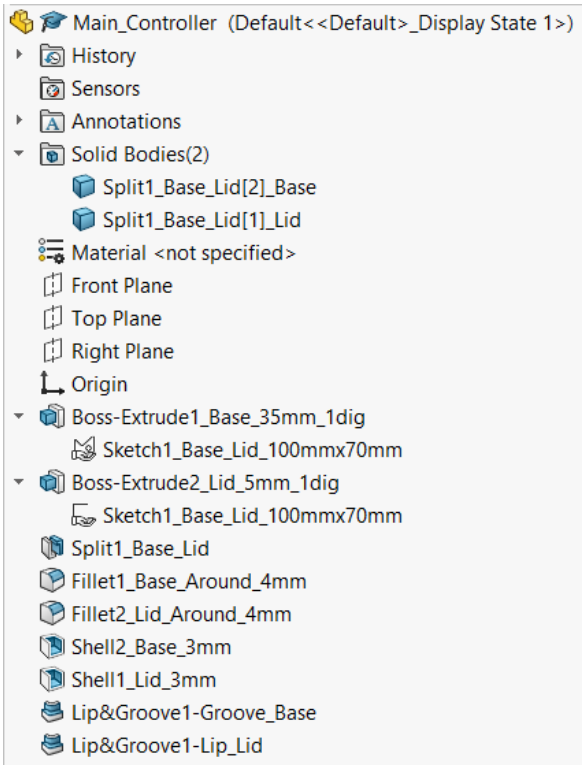


Figure E.1: Main Controller Hierarchy Part 1

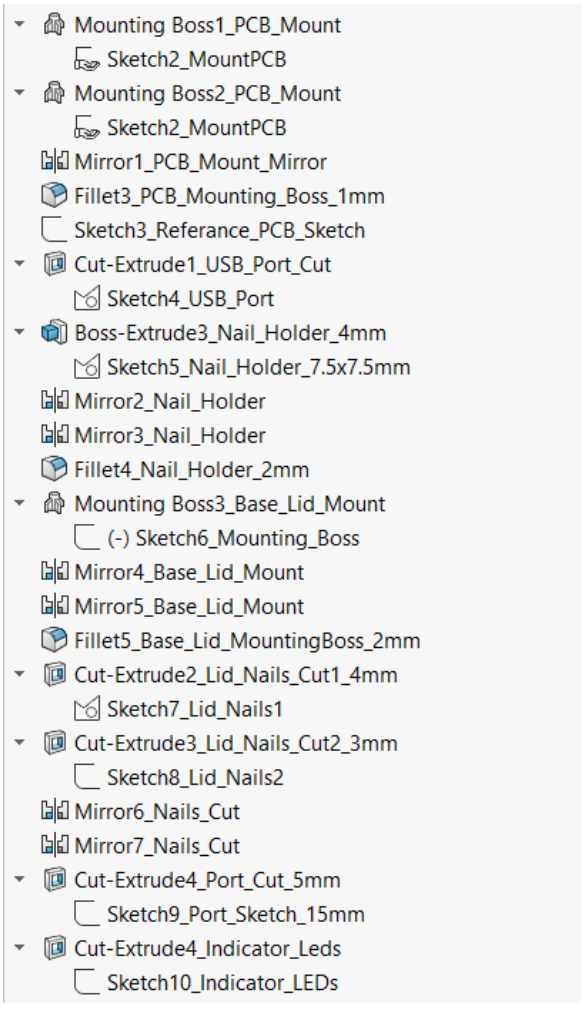


Figure E.2: Main Controller Hierarchy Part 2

Module

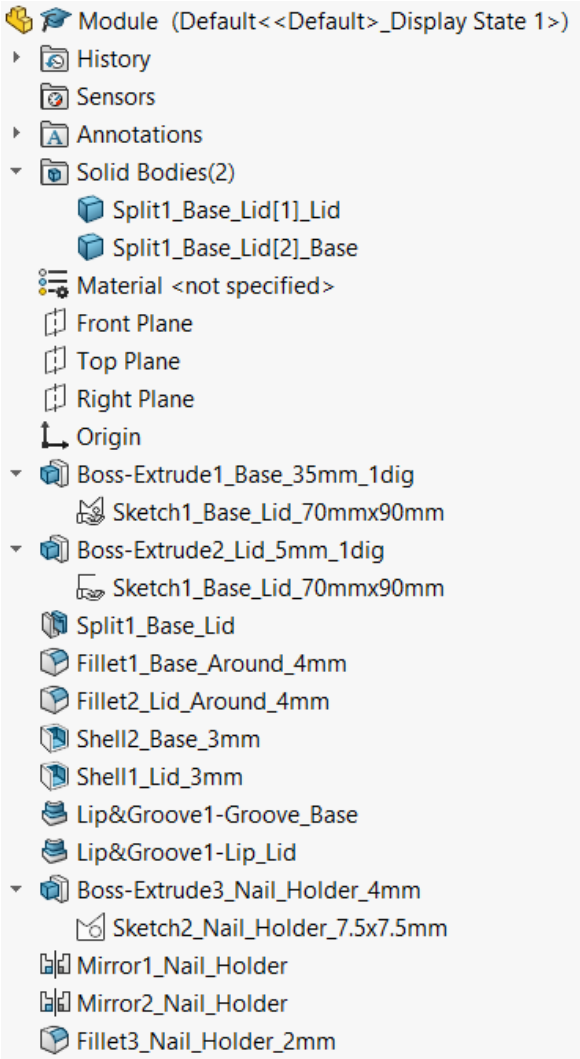


Figure E.3: Module Hierarchy Part 1

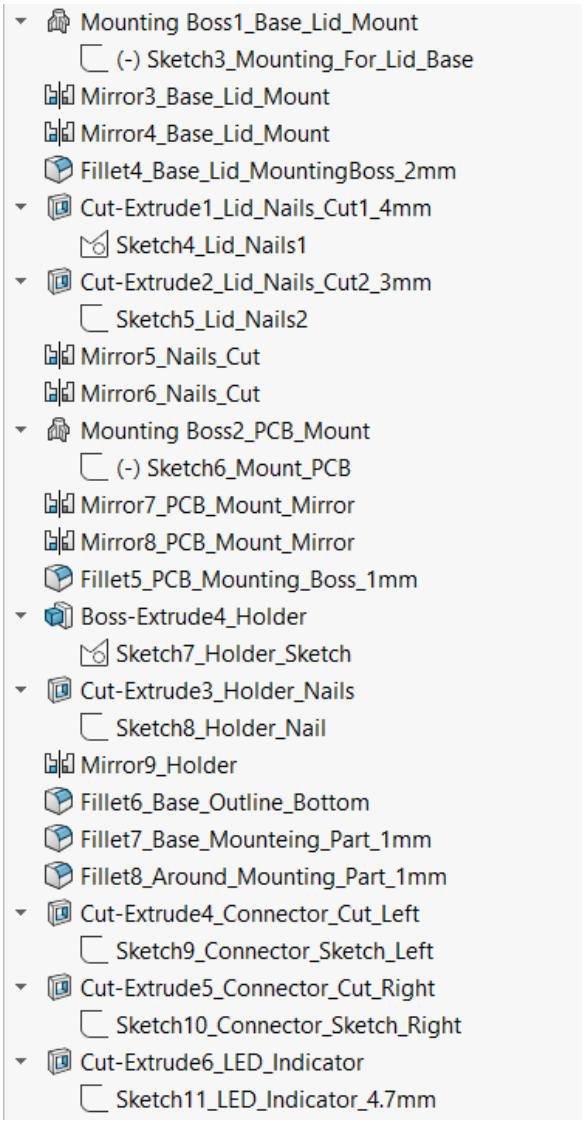


Figure E.4: Module Hierarchy Part 2

With

1. Well-structured Design
2. Fully defined Sketches

sketches we were able to do a standard enclosure design

E.2 Draft Analysis

Main Controller

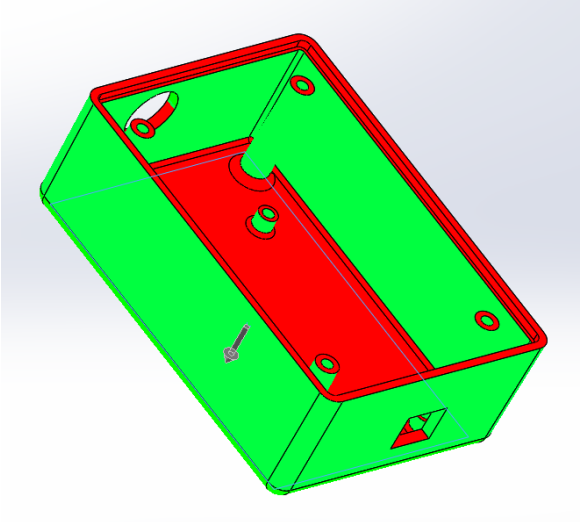


Figure E.5: Main Controller Base Draft Analysis

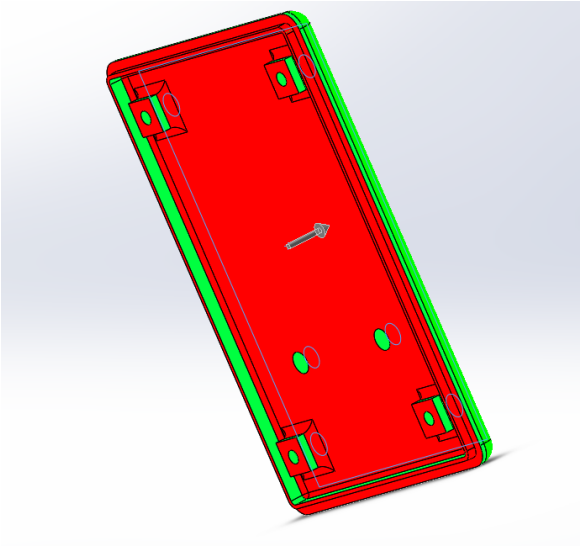


Figure E.6: Main Controller Lid Draft Analysis

Module

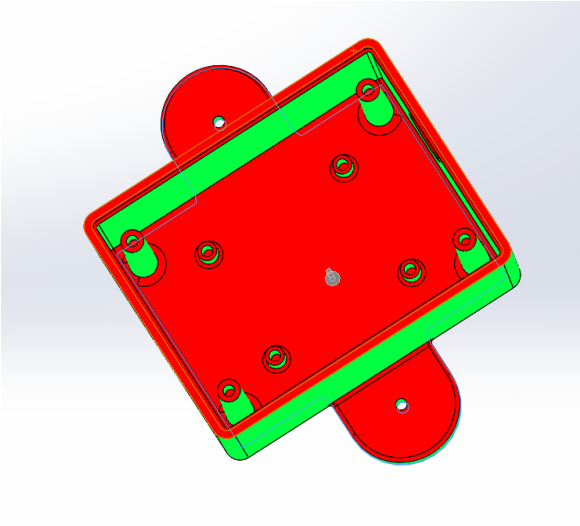


Figure E.7: Module Base Draft Analysis

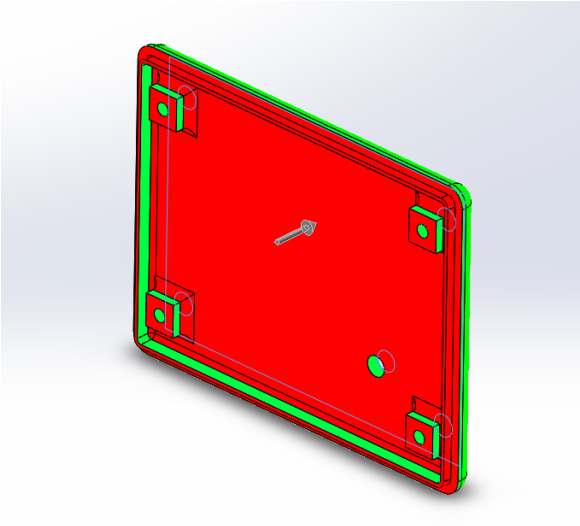


Figure E.8: Module Lid Draft Analysis

E.3 Photographs of Final Product

Here are some photographs of the final printed enclosure.

Base

Figure E.9: Base Top

Figure E.10: Base Lid Bottom

Figure E.11: Base USB Port

Module

Figure E.12: Module Top

Figure E.13: Module Front

Figure E.14: Module Top

Appendix F

Codes related to Software and Firmware

Visit our GitHub organization `https://github.com/VibroGuard`

Figure F.1: QR for GitHub

Appendix G

Website

Visit Our Website for more details `http://vibroguard.unaux.com/`

Figure G.1: QR for Website

More details available on the website

- Product Overview
- All the documents related to the project
- Executable file of the application

Bibliography

[1] "Atmel AVR Hardware Design," [Online]. Available: https://ww1.microchip.com/downloads/en/appnotes/atmel-2521-avr-hardware-design-considerations_applicationnote_avr042.pdf.

[2] "ATmega328P Datasheet," [Online]. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf.

[3] "FTDI FT232RL Breakout Schematic," [Online]. Available: https://www.sparkfun.com/datasheets/BreakoutBoards/FT232RL-Breakout-Schematic.pdf.

[4] "YouTube - Introduction to Anomaly Detection for Engineers," [Online]. Available: https://www.youtube.com/watch?v=1TJZUERSSUo.

[5] "YouTube - LSTM AutoEncoders for Anomaly Detection," [Online]. Available: https://www.youtube.com/watch?v=6S2v7G-OupA.

[6] "LSTM AutoEncoder Anomaly Detection," [Online]. Available: https://github.com/bnsreenu/python_for_microscopists/blob/master/180_LSTM_encoder_decoder_anomaly_GE.py.

[7] "Is using the CH340G on a commercial product a horrible idea: https://electronics.stackexchange.com/questions/440524/is-using-the-ch340g-on-a-commercial-product-a-horrible-idea

[8] https://matplotlib.org/

[9] https://docs.python.org/3/library/tkinter.html

[10] https://cloud.arduino.cc/, https://azure.microsoft.com/en-gb/

[11] https://en.wikipedia.org/wiki/CAN_bus

[12] https://www.mouser.com/catalog/specsheets/xs2_ds_csm1342.pdf

[13] https://www.omron-ap.co.nz/products/family/782/dimension.html

[14] https://www.mouser.com/

[15] https://jlcpcb.com/

Declaration of Review

Reviewed by: Sajitha Madugalle 210351M

Group: Group 28 K

Notes:

The report is easy to understand. Defining abrevation at the begining to write the report is appriciated.
Added draft analysis for describe moldability and manufacturbility

The project was explained graphicaly by block diagrams.

Better if the dimentions of the designed PCB are mentioned.

I hereby declare that I have reviewed the design documentation for the project **Industrial Machine Vibrations Monitoring System**, by the **Group: 30 (K)**, with thorough inspection of the schematic and mold designs. I confirm that the comments made by above are my own and reflects unbiased critiques of the device design methodology.

23/06/2024

Date **Signature**

Declaration of Review

Reviewed by: Liyanaarachchi L.A.S.

Group: Group K (29)

Notes:

Design Documentation for Industrial Machine Vibrations Monitoring System is well-organized & detailed document. The PCB part & enclosure design are very well detailed. And also PCB and enclosure are pretty good.
Overall the documentation is well-structured.

I hereby declare that I have reviewed the design documentation for the project **Industrial Machine Vibrations Monitoring System**, by the **Group: 30 (K)**, with thorough inspection of the schematic and mold designs. I confirm that the comments made by above are my own and reflects unbiased critiques of the device design methodology.

15/06/2024

Date

Signature